

AD-A060 941

TEXAS TECH UNIV LUBBOCK INST FOR ELECTRONICS SCIENCE
FAULT ANALYSIS IN ELECTRONIC CIRCUITS AND SYSTEMS. II.(U)
JAN 78 R SAEKS, N SEN, H M CHEN, K S LU

F/G 9/3

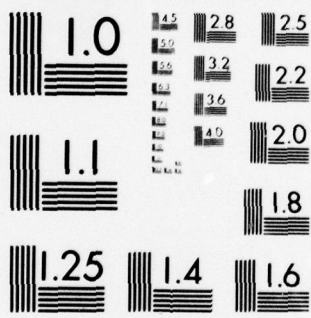
N00014-75-C-0924

UNCLASSIFIED

NL

1 OF 4
AD
A060941





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD A060941

DDC FILE COPY

LEVEL III

71 Ho18633

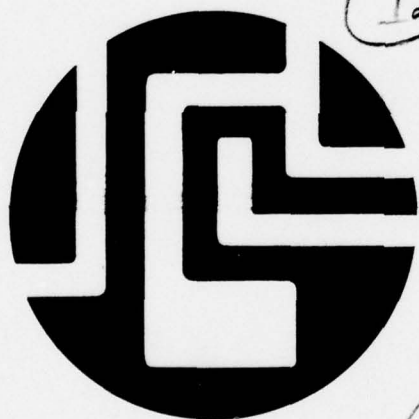
6 Fault Analysis in Electronic Circuits and Systems. II.

10 R. Saeks, N. Sen, H. M. S. Chen, K. S. Lu
S. Sangani and R. A. DeCarlo

9 Interim rept.

12 359p.

11 Jan 78



15 N00014-75-C-0924,
N00014-76-C-1136

Use →

Institute for
Electronics Science

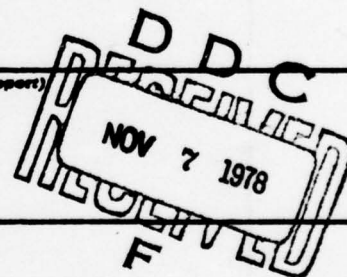
This document has been approved
for public release and sale; its
distribution is unlimited.

TEXAS TECH UNIVERSITY
Lubbock, Texas 79409

78 0.8 01 027
410 354

JOB

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Fault Analysis in Electronic Circuits and Systems II		5. TYPE OF REPORT & PERIOD COVERED Interim
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) R. Saeks, N. Sen, H.M.S. Chen, K.S. Lu, S. Sangani and R.A. DeCarlo		8. CONTRACT OR GRANT NUMBER(s) N00014-75-C-0924✓ N00014-76-C-1136 NR-048-619
9. PERFORMING ORGANIZATION NAME AND ADDRESS Texas Tech University Dept. of Electrical Engineering Lubbock, Texas 79409		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 122105 121409
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research 800 N. Quincy St. Arlington, Virginia 22217		12. REPORT DATE January 1978
		13. NUMBER OF PAGES 354 + iii
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for Public Release, distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Fault Analysis, Fault Prediction, Fault Diagnosis, Fault Isolation, Testability, Test Points, Test Frequencies		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) A summary of several research projects in the fault analysis area. Specific topics include test point sections, the formulation of a measure of testability for analog electronic circuits, test frequency selection, and fault diagnosis in linear and affine circuits.		



FAULT ANALYSIS IN ELECTRONIC CIRCUITS AND SYSTEMS II*

R. Saeks, N. Sen, H.S.M. Chen, K.S. Lu,
S. Sangani, and R.A. DeCarlo

ACCESSION for	
NTIS	White Section <input checked="" type="checkbox"/>
DDC	Buff Section <input type="checkbox"/>
UNANNOUNCED	<input type="checkbox"/>
JUSTIFICATION	
BY	
DISTRIBUTION/AVAILABILITY CODES	
D.	SPECIAL
A	

*This research supported in part by Office of Naval Research contracts 75-C-0924 and 76-C-1136.

FORMULATION FO THE FAULT DIAGNOSIS EQUATIONS*

N. Sen** and R. Saeks

*This research supported in part by Office of Naval Research Contracts 75-C-0924 and 76-C-1136.

**Presently with the Datapoint Corporation, San Antonio, Texas.

TABLE OF CONTENTS

	Page
Formulation of the Fault Diagnosis Equation----- N. Sen and R. Saeks	1
A Search Algorithm for the Solution of the Fault Diagnosis Equations----- H.M.S. Chen and R. Saeks	25
An Approach to Built-In Testing----- R. Saeks	57
Failure Prediction for an On-Line Maintenance System in a Poisson Shock Environment----- K.S. Lu and R. Saeks	81
Fault Isolation Via Affinization----- R. Saeks, R.A. DeCarlo and S. Sangani	109
Fault Analysis in Affine Sequential Circuits----- S. Sangani and R. Saeks	117
A Measure of Testability and Its Application To Test Point Selection - Theory----- N. Sen and R. Saeks	125
A Measure of Testability and Its Application To Test Point Selection - Computation----- N. Sen and R. Saeks	135
A Measure of Testability and Its Application To Test Point Selection ----- N. Sen	145
A Search Algorithm for the Solution of the Fault Diagnosis Equations----- H.M.S. Chen	251
ECMP - A Software Package for Computation of Measurability and Cost To Aid Test Point Selection----- N. Sen	299

78 08 01 027

Abstract

The fault diagnosis problem for a linear system whose transfer function matrix is measured at a discrete set of frequencies is formalized. A measure of solvability for the resultant equations and a measure of testability for the unit under test is developed. These, in turn, are used as the basis of algorithms for choosing test points and test frequencies.

I. Introduction

Conceptually, the fault analysis problem for an analog circuit or system amounts to the measurement of a set of externally accessible parameters of the system from which one desires to determine the internal system parameters or equivalently locate the failed components as illustrated in Figure 1.

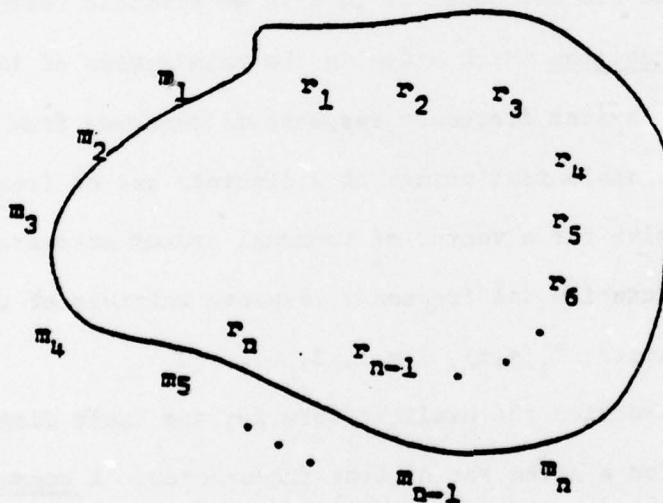


Figure 1. Conceptual Model of Fault Diagnosis Problem.

Here, the measurements, m_i , may represent data taken at distinct test points or alternatively, data taken at a fixed test point under different stimuli. Similarly, the r_i represent parameters characterizing the various internal system components. Here, a single parameter may characterize an entire component, say a resistance, capacitance or inductance. Alternatively, a component may be

represented by several parameters: the h-parameters of a transistor, the poles and gain of an op-amp, etc. In general, one models a system component by the minimum number of parameters which will allow the failure to be isolated up to a shop replaceable assembly (SRA) with all "allowed" system failures manifesting themselves in the form of some parameter change.

To solve the fault diagnosis problem, one then measures $m = \text{col}(m_1)$ and solves a nonlinear algebraic equation

$$1. \quad m = F(r)$$

for $r = \text{col}(r_1)$ to diagnose the fault. The parameters in the resultant r vector which are out of tolerance then indicate the faulty component.⁶

The purpose of the present paper is to give an explicit formulation of the fault diagnosis equations which arise in the maintenance of linear systems. Here, one measures the system frequency response as observed from a specified set of externally accessible test points at a discrete set of frequencies and it is desired to solve for a vector of internal system parameters, r , which completely characterize the frequency response matrices of the individual system components; $Z_i(s, r)$, $i = 1, 2, \dots, q$.

In the following section the explicit form for the fault diagnosis equations is derived for a given set of test frequencies. A measure of solvability of these equations is then developed in section 3 and employed in section 4 in an algorithm for optimally selecting test frequencies. The measure of solvability for the fault analysis equations, given an optimal choice of test frequencies, is then taken as a measure of testability^{1,2,5} for the unit under test (UUT) and is used as the basis of an algorithm for the optimal choice of test points.^{3,4,5} Finally, a number of examples are presented in section 5.

II. Explicit Form of the Fault Diagnosis Equations

In the case of a linear time-invariant circuit or system, the fault diagnosis equations may be expressed in analytical form.⁶ Since the fault diagnosis equations deal with the relationship between the externally measurable system parameters, u , and the internal component parameters, r , we adopt a component connection model as the starting point for the derivation of the fault diagnosis equations.^{7,8} This is one of several commonly employed large scale system models in which the components and connections in a circuit or system are modeled by distinct equations, thereby permitting one to explicitly deal with the relationship between the individual component parameters and the composite system parameters.

Since the present study is restricted to linear time-invariant systems, we assume that each component is characterized by a transfer function matrix which is dependent on the potentially variable component parameters, $Z_i(s, r)$. For the classical RLC components $Z_i(s, r)$ may take the form R , Ls , or $1/sC$ for the case of a resistor, inductor, or capacitor, respectively. More generally, one may model an op-amp by the transfer function $k/(s-p_1)(s-p_2)$ where the parameter vector, r , now represents the three potentially variable component parameters; k , p_1 , p_2 ; or a delay by ke^{sT} , etc. Although the symbol Z is used, the components are not assumed to be represented by impedance matrices. Indeed, hybrid models are used in most of our examples. For the purpose of analysis, it is assumed that all faults manifest themselves in the form of changes, possibly catastrophic, in the parameter vector, r , with the frequency characteristics of the components unchanged. Although not universal, this fault hypothesis covers the most commonly encountered situations and subsumes the common industrial practice of assuming that all failures in analog circuits and systems take the form of open and short circuited components.⁹

Our system components are thus characterized by a set of simultaneous equations

$$2. \quad b_i = Z_i(s,r)a_i \quad i = 1, 2, \dots, q$$

where a_i and b_i denote the component input and output vectors, respectively. For notational brevity, these component equations may be combined into a single block diagonal matrix equation

$$3. \quad b = Z(s,r)a$$

where $b = \text{col}(b_i)$, $a = \text{col}(a_i)$ and $Z(s,r) = \text{diag}(Z_i(s,r))$.

Although there are many ways to represent the connection in a circuit or system; say a block diagram, linear graph or signal flow graph, any such representation is simply a graphical means for displaying a set of connection equations: Kirchoff laws, adder equations, etc. As such, for our component connection model we adopt a purely algebraic connection model in which the connection equations are displayed explicitly without the intermediary of some kind of graphical connection diagram. This takes the form

$$4. \quad \begin{aligned} a &= L_{11}b + L_{12}u \\ y &= L_{21}b + L_{22}u \end{aligned}$$

where u and y represent the vectors of accessible inputs and outputs which are available to the test system. In simple systems, the connection matrices, L_{ij} , are usually obtainable by inspection, whereas, in more complex systems, computer codes have been developed for their derivation.⁷ Moreover, they are assured to exist in all but the most pathological systems.⁸

It is the pair of simultaneous matrix equations 3 and 4 which are termed the component connection model. By combining equations 3 and 4 to eliminate the component input and output variables, a and b , one may derive^{6,7} an expression for the transfer function matrix observable by the test system between the test

input and output vectors, u and y , obtaining

$$5. \quad S(s,r) = L_{22} + L_{21}(1 - Z(s,r)L_{11})^{-1}Z(s,r)L_{12}$$

where

$$6. \quad y = S(s,r)u$$

For a linear time-invariant system the transfer function $S(s,r)$ is a complete description of the measurable data about the unit under test available to the test system. Moreover, being rational it is completely determined by its value at a finite number of frequencies. As such, without loss of generality, we may take our measured data to be of the form

$$7. \quad \text{col}[S(s_1,r), S(s_2,r), \dots, S(s_k,r)]$$

The fault diagnosis equations then take the form

$$8. \quad \begin{bmatrix} S(s_1,r) \\ S(s_2,r) \\ \cdot \\ \cdot \\ \cdot \\ S(s_k,r) \end{bmatrix} = \begin{bmatrix} L_{22} + L_{21}(1-Z(s_1,r)L_{11})^{-1}Z(s_1,r)L_{12} \\ L_{22} + L_{21}(1-Z(s_2,r)L_{11})^{-1}Z(s_2,r)L_{12} \\ \cdot \\ \cdot \\ \cdot \\ L_{22} + L_{21}(1-Z(s_k,r)L_{11})^{-1}Z(s_k,r)L_{12} \end{bmatrix}$$

Since $S(s,r)$ is, in general, a matrix, the fault diagnosis equations as derived above take the form of a matrix $(\text{col}[S(s_i,r)])$ valued function of a vector valued variable, r . Computationally, however, we prefer to work with a vector valued function of a vector valued variable and hence, we transform $S(s,r)$ into a column vector via

$$9. \quad \text{vec}[S(s,r)] = \text{Col } [S^i(s,r)]$$

where $S^i(s,r)$ denotes the i th column of the matrix, $S(s,r)$. With the aid of the identity $\text{vec}[XYZ] = [Z^T \otimes X] \text{vec } [Y]$ equation 8. then transforms into^{7,12}

$$10. \quad M = \begin{bmatrix} \text{vec}[S(s_1,r)] \\ \text{vec}[S(s_2,r)] \\ \vdots \\ \text{vec}[S(s_k,r)] \end{bmatrix} = \begin{bmatrix} \text{vec}[L_{22}] + [L_{12}^T \otimes L_{21}(1-Z(s_1,r)L_{11})^{-1}] \text{vec}[Z(s_1,r)] \\ \text{vec}[L_{22}] + [L_{12}^T \otimes L_{21}(1-Z(s_2,r)L_{11})^{-1}] \text{vec}[Z(s_2,r)] \\ \vdots \\ \text{vec}[L_{22}] + [L_{12}^T \otimes L_{21}(1-Z(s_k,r)L_{11})^{-1}] \text{vec}[Z(s_k,r)] \end{bmatrix} = F(r)$$

which is the form of the fault diagnosis equations with which we desire to work.

III. Solvability of the Fault Diagnosis Equations

For the fault diagnosis equations derived above to be a viable tool of circuit and system diagnosis two fundamental questions remain to be answered: "What test frequencies should be employed to optimize the solvability of the equations?" and "How solvable are the equations given an optimal choice of test frequencies?" Both of these questions, in turn, hinge on the development of some type of measure of solvability for the fault diagnosis equations.

For a set of linear equations

$$11. \quad m = Fr$$

where r is an n -vector, m is a p -vector and F is a p by n matrix one may characterize the solvability of the equations in terms of the number of arbitrary parameters in its solution (if a solution exists). As such, $\delta = n - \text{rank}(F)$ is a natural measure of the solvability for equation 11. Here, $\delta = 0$ implies that the equation has a unique solution, $\delta = 1$ implies that the solution is determined up

to one arbitrary parameter and so on, with increasing values of δ representing decreasing degrees of solvability.

Unfortunately, the fault diagnosis equations are nonlinear even for linear systems and hence we must resort to the implicit function theorem to obtain a measure of solvability analogous to the above.¹³ Indeed, if r_f is a solution to the fault diagnosis equations, then r_f is determined up to a

$$12. \quad \delta(r_f) = n - \text{rank} \left[\left[\frac{dF}{dr} (r_f) \right] \right]$$

dimensional manifold (of arbitrary parameters) in a neighborhood of r_f . Here dF/dr is the Jacobian matrix of partial derivatives of F with respect to r . With the aid of the matrix identity $d(M^{-1})/dr = -M^{-1}[dM/dr]M^{-1}$, dF/dr can be computed explicitly from equations 8. and 10. yielding

$$13. \quad \frac{dF}{dr}(r_f) = \begin{bmatrix} \{ ([1+L_{11}(1-Z(s_1, r_f)L_{11})^{-1}Z(s_1, r_f)]L_{12})^t \otimes (L_{21}(1-Z(s_1, r_f)L_{11})^{-1})^h [(d\text{vec } Z(s_1, r_f))/dr] \\ \{ ([1+L_{11}(1-Z(s_2, r_f)L_{11})^{-1}Z(s_2, r_f)]L_{12})^t \otimes (L_{21}(1-Z(s_2, r_f)L_{11})^{-1})^h [(d\text{vec } Z(s_2, r_f))/dr] \\ \vdots \\ \{ ([1+L_{11}(1-Z(s_k, r_f)L_{11})^{-1}Z(s_k, r_f)]L_{12})^t \otimes (L_{21}(1-Z(s_k, r_f)L_{11})^{-1})^h [(d\text{vec } Z(s_k, r_f))/dr] \end{bmatrix}$$

where "t" denotes matrix transposition and \otimes denotes the matrix Kronecker (or tensor) product.

The difficulty with the implicit function theorem is that it only yields local information valid in a neighborhood of a solution. Fortunately, however, given the special nature of the Jacobian matrix of equation 13. coupled with an assumption that the component transfer function matrices $Z_i(s, r)$ are rational in r , it is possible to show that the rank of the Jacobian matrix is "almost constant." This,

in turn, allows us to transform the local measure of solvability of equation 12. into a global measure of solvability. For this purpose we adopt the algebraic geometric definition for the term "almost constant." I.e. we say that a function of r_f is almost constant if it is constant except possibly for those values of r_f lying in an algebraic variety (the solution space of a finite set of non-zero simultaneous polynomial equations in n variables). More generally, we say that a property holds "almost everywhere" (a.e.) or for almost all r_f in n -space if it is true for all values of r_f except possibly those lying in an algebraic variety. Since the Lebesgue measure of an algebraic variety is zero, this definition for the concept "almost everywhere" is consistent with the more common measure theoretic definition and is more natural in the context of our application.¹⁴

Theorem 1: Let $Z_i(s, r)$; $i = 1, 2, \dots, q$; be rational in r . Then $\delta(r_f)$ is almost constant.

Note, the assumption that $Z_i(s, r)$ is rational in r is quite minor being satisfied by all of the examples given in section II except for the delay (which can be approximated by a function which is rational in r). In practice, the component transfer function matrices will also be rational in s though this is not required for the present theorem since F and dF/dr are formulated in terms of specific test frequencies, s_1, s_2, \dots, s_k . Given our assumption on the $Z_i(s, r)$, together with equation 13., it then follows that $\frac{dF}{dr}(r_f)$ is also rational in r_f .

Proof of Theorem 1: We begin by showing that an arbitrary polynomial matrix in r , $P(r)$, has almost constant rank. Since rank $P(r)$ is restricted to the finite set of integers $(0, 1, 2, \dots, j)$; where j is the minimum of the number of rows and columns in $P(r)$, there exists an r_m which maximizes the rank of $P(r)$

$$14. \quad \text{rank}[P(r_m)] \geq \text{rank}[P(r)]$$

Now, the rank of a matrix is the dimension of its largest non-singular square sub-matrix. As such, $P(r)$ admits a square sub-matrix, $M(r)$, whose dimension is

equal to the rank $P(r_m)$ and for which

$$15. \quad \det M(r_m) \neq 0.$$

Now, $\det[M(r)]$ is a polynomial in r which is not identically zero (from equation 15.) and hence, it is non-zero a.e. As such,

$$16. \quad \text{rank}[P(r_m)] \geq \text{rank}[P(r)] \geq \text{rank}[M(r)] = \text{rank}[P(r_m)] \text{ a.e.}$$

showing that $\text{rank}[P(r)] = \text{rank}[P(r_m)]$ almost everywhere. As such, $\text{rank}[P(r)]$ is almost constant.

Now, to verify that $\text{rank} \left[\frac{dF}{dr}(r_f) \right]$ is constant we decompose this matrix as

$$17. \quad \frac{dF}{dr}(r_f) = \frac{P(r_f)}{d(r_f)}$$

where $P(r_f)$ is a polynomial matrix and $d(r_f)$ is a non-zero common denominator. $P(r_f)$ has almost constant rank while $d(r_f)$ is non-zero almost everywhere and hence can effect the rank of $P(r_f)$ only on an algebraic variety (since the division of a matrix by a non-zero scalar does not effect its rank.) As such, our Jacobian matrix has almost constant rank implying that

$$18. \quad \delta(r_f) = n - \text{rank} \left[\frac{dF}{dr}(r_f) \right]$$

is also almost constant. The proof of the Theorem is therefore complete.

Given the theorem, we may now define a global measure of solvability for the fault diagnosis equation, δ , as the generic value of $\delta(r_f)$. I.e. the value $\delta(r_f)$ takes on for almost all r_f . This proves to be a natural measure of solvability since it indicates the ambiguity which will result from an attempt to solve the fault diagnosis equations in a neighborhood of almost any failures. Of course, one requires some sort of equation solving algorithm^{10,11} to locate a neighborhood of an actual failure. The δ parameter, however, represents a bound on the

performance of any such algorithm. Finally, we note that since δ is independent of r_f , the solution of the fault diagnosis equations, it can be computed at the time the system and its test algorithm are developed by evaluating $\delta(r)$ at a randomly chosen generic point, say r_0 . In turn, this parameter may then be employed as an aid in the choice of test frequencies and test points.

IV. Test Frequency Selection

Adopting the measure of solvability, δ , formulated in the preceding section, it remains to develop an algorithm for choosing a set of test frequencies; s_1, s_2, \dots, s_k ; which maximize the solvability of the fault diagnosis equations (i.e. minimize δ). To this end, let δ_{\min} denote the minimum value achieved by δ for any set of test frequencies; s_1, s_2, \dots, s_k ; $k = 1, 2, \dots$. Since the possible values for δ are restricted to the finite set; $\delta = 0, 1, \dots, n$; such a minimum is assured to exist.

The following theorem gives an explicit formula for computing δ_{\min} while its proof yields an algorithm for choosing a set of test points which achieve δ_{\min} . Since the purpose of this theorem is to formulate an algorithm for choosing test frequencies, the theorem is expressed in terms of

$$19. \quad \text{vec}[S(s,r)] = \text{vec}[L_{22}] + [L_{12}^t] \otimes L_{21}(1-Z(s,r)L_{11})^{-1} \text{vec}[Z(s,r)]$$

and

$$20. \quad \frac{d\text{vec}[S(s,r)]}{dr} = \{ [(1 + L_{11}(1-Z(s,r)L_{11})^{-1}Z(s,r)]L_{12}^t \otimes (L_{21}(1-Z(s,r)L_{11})^{-1}) \} \\ \{ d\text{vec}[Z(s,r)]/dr \}$$

viewed as rational functions in s rather than in terms of the function $F(r)$ which is formulated in terms of an a-priori choice of test frequencies.

Theorem 2: Let $Z_i(s,r)$; $i = 1, 2, \dots, q$; be rational in s and r . Then

$$\delta_{\min} = n - \text{col-rank} \left[\frac{d\text{vec}[S(s,r)]}{dr} \right]$$

where n is the dimension of the parameter vector, r , and "col-rank" denotes the generic number of linearly independent columns of the rational matrix $[\text{dev}[S(s,r)]/\text{dr}]$ over the field of complex numbers. Moreover, δ_{\min} is achieved by almost any choice of $n - \delta_{\min}$ distinct complex frequencies.

Proof: For the sake of brevity, we will prove the theorem only for the special case where $S(s,r)$ is a scalar transfer function (allowing us to drop the "vec" transformation) though essentially the same proof goes through in the general case modulo some notational complexities.⁵ Also, since the rank of the Jacobian matrix is almost constant it suffices to fix the parameter vector, r , at any generic point, say r_0 . This then reduces $[\text{dvec}[S(s,r)]/\text{dr}]$ to a row vector of rational functions

$$21. \quad R(s) = [R_1(s) \ R_2(s) \ \dots \ R_n(s)]$$

where

$$22. \quad R_i(s) = [\text{dvec}[S(s, r_0)]/\text{dr}_i]$$

and our problem reduces to the verifications of the fact that the number of linearly independent columns of $R(s)$ over the field of complex scalars is equal to the maximum possible rank of the complex matrix

$$23. \quad \begin{bmatrix} R(s_1) \\ R(s_2) \\ \vdots \\ R(s_k) \end{bmatrix} = \begin{bmatrix} R_1(s_1) & R_2(s_1) & \dots & R_n(s_1) \\ R_1(s_2) & R_2(s_2) & & R_n(s_2) \\ \vdots & \vdots & & \vdots \\ R_1(s_k) & R_2(s_k) & \dots & R_n(s_k) \end{bmatrix} = \text{col}(R(s_i))$$

over all possible choices of the complex frequencies; s_1, s_2, \dots, s_k ; $k = 1, 2, \dots$.

Now, clearly if some column of $R(s)$, say the n th, is dependent on the remaining columns, then

$$24. \quad R_n(s) = \sum_{j=1}^{n-1} c_j R_j(s)$$

for all s . Then by applying 24. individually for each s_i

$$25. \quad \text{col}(R_n(s_i)) = \sum_{j=1}^{n-1} c_j \text{col}(R_j(s_i))$$

for any possible number or choice of the s_i . The rank of the matrix of Equation 23 is therefore less than or equal to the number of linearly independent columns of $R(s)$ over the field of complex numbers.

To prove that equality can be achieved with an appropriate choice of $n - \delta_{\min}$ complex test frequencies, s_i , we invoke our assumption that $S(s, r)$ is a scalar transfer function. Without loss of generality, we may assume that $R_1(s)$ through $R_q(s)$ are the linearly independent entries in $R(s)$ over the field of complex numbers in which case we must show that there exists complex frequencies s_1, s_2, \dots, s_k ($k = q$ in this case) which make the first q columns of the matrix of equation 23. linearly independent.

If $q = 1$, $R_1(s)$ is not identically zero (since otherwise it would be linearly dependent) and hence for almost all s_1 , $R_1(s_1) \neq 0$. As such, the columns in this trivial one by one matrix are linearly independent. With this as a starting point, we will use an inductive argument to show that the theorem holds for all values of q . We, therefore, assume that it has been shown that for $q = p$ there exist complex frequencies; s_1, s_2, \dots, s_p ; such that the matrix

$$26. \quad R_p = \begin{bmatrix} R_1(s_1) & R_2(s_1) & \dots & R_p(s_1) \\ R_1(s_2) & R_2(s_2) & \dots & R_p(s_2) \\ \vdots & \vdots & \ddots & \vdots \\ R_1(s_p) & R_2(s_p) & \dots & R_p(s_p) \end{bmatrix}$$

has linearly independent columns and we desire to show that there exists an s_{p+1} such that the matrix

$$27. \quad \underline{R}_{p+1}(s) = \begin{bmatrix} R_1(s_1) & R_2(s_1) & \dots & R_p(s_1) & R_{p+1}(s_1) \\ R_1(s_2) & R_2(s_2) & \dots & R_p(s_2) & R_{p+1}(s_2) \\ \vdots & \vdots & & \vdots & \vdots \\ R_1(s_p) & R_2(s_p) & \dots & R_p(s_p) & R_{p+1}(s_p) \\ R_1(s) & R_2(s) & & R_p(s) & R_{p+1}(s) \end{bmatrix}$$

has linearly independent columns for $s = s_{p+1}$. By virtue of our assumption that $S(s,r)$ is a scalar both \underline{R}_p and $\underline{R}_{p+1}(s)$ are square and we may test for linear independence of the columns of $\underline{R}_{p+1}(s)$ by computing its determinant. Expanding 27. in co-factors along its bottom row, we obtain

$$28. \quad \det(\underline{R}_{p+1}(s)) = \sum_{j=1}^{p+1} (-1)^{p+j+1} \Delta_{p+1, j} R_j(s)$$

Since \underline{R}_p has linearly independent columns $\Delta_{p+1, p+1} \neq 0$, hence, the coefficients in the summation of equation 28. are not all zero and thus by the linear independence of the $R_i(s)$ the summation is not identically zero. As such, one can choose almost any s_{p+1} which will make the determinant of $\underline{R}_{p+1}(s_{p+1})$ non-zero thus assuring the \underline{R}_{p+1} has linearly independent columns when its rows are evaluated at the complex frequencies s_1, s_2, \dots, s_{p+1} . The proof of the theorem is thus complete.

Note that the proof of the theorem yields a natural sequential algorithm for choosing test frequencies. Moreover, for the scalar case we have shown that the number of required test frequencies is exactly $n - \delta_{\min}$ (equal to the column rank of the Jacobian matrix). In the general case where $S(s,r)$ is not a scalar, the number of required test frequencies is less than or equal to $n - \delta_{\min}$.⁵

Although the theorem implies that one can randomly choose almost any $n - \delta_{\min}$ test frequencies to maximize the solvability of the fault diagnosis equations, the result does not take cognizance of numerical considerations. Although no theory yet exists for choosing test points with numerical considerations in mind, it has been our experience that the "well posedness" of the fault diagnosis equations is quite sensitive to the choice of test frequencies.⁵ In most of our experiments, we have worked with real test frequencies to eliminate the necessity of working in the complex plane. On the other hand, m is most easily measured when values of s_i on the $j\omega$ axis are employed whereas it has been suggested that test frequencies symmetrically spaced around a circle in the complex plane might yield numerically "well posed" equations.

Although the measure of solvability, δ , for the fault diagnosis equations is dependent on the choice of test frequencies, as well as the properties of the unit under test, δ_{\min} is determined entirely by the UUT; its components, connections and accessible test points; and is completely independent of the test algorithm employed. As such, δ_{\min} may be taken as a natural measure of testability¹ for the UUT which characterizes the degree to which the fault analysis equations can be solved given an optimal choice of test frequencies and solution algorithm. Moreover, δ_{\min} may be used as an aid for the optimal selection of test points.^{3,4,5} To this end we may choose a set of test points, from several options, so as to minimize δ_{\min} . Alternatively, we may attribute a cost to each input and output test point and then choose the least cost combination of test points which yield a specified δ_{\min} . This latter process reduces to a rather straightforward integer programming problem and is thus readily automated.^{4,5} The technique is illustrated in the examples of the following section.

V. Examples

An initial illustration of the theory consider the RC coupled amplifier with inductive load shown in Figure 2. Here we will take E_i to be the only

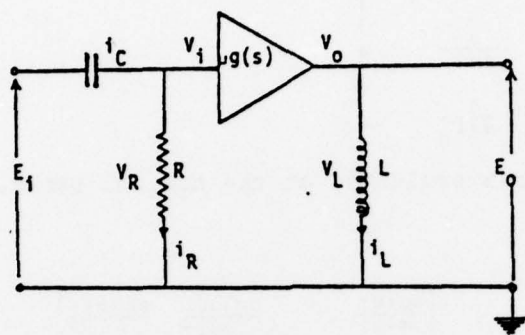


Figure 2: RC coupled amplifier with inductive load.

test input but we will initially allow E_o , i_L , i_C , and V_i to all be taken as test outputs with the measure of testability, δ_{\min} , being used to extract a reduced set of test outputs from these options. A component connection model for this circuit is given by

$$28. \quad \begin{bmatrix} v_o \\ i_L \\ v_c \\ i_R \end{bmatrix} = \begin{bmatrix} U_g(s) & 0 & 0 & 0 \\ 0 & 1/LS & 0 & 0 \\ 0 & 0 & 1/CS & 0 \\ 0 & 0 & 0 & 1/R \end{bmatrix} \begin{bmatrix} v_i \\ v_L \\ i_C \\ v_R \end{bmatrix}$$

and

$$29. \quad \begin{bmatrix} v_i \\ v_L \\ i_C \\ v_R \\ \vdots \\ E_o \\ i_o \\ i_R \\ v_i \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_o \\ i_L \\ v_c \\ i_R \\ \vdots \\ E_i \end{bmatrix}$$

Taking our vector of potentially variable component parameters to be

$r = \text{col}(\mu, L, C, R)$ each with unity nominal value, we obtain a nominal trans-

fer function matrix

$$30. \quad S(s, r) = \begin{bmatrix} \frac{s(q(s)+1) + 1}{s+1} \\ \frac{q(s)}{s+1} \\ \frac{s}{s+1} \\ \frac{s}{s+1} \end{bmatrix}$$

whereas our Jacobain matrix evaluated at the nominal parameter values is given by

$$31. \quad \frac{d\text{vec}[S(s, r)]}{dr} = \begin{bmatrix} \frac{sq(s)}{s+1} & 0 & \frac{sq(s)}{(s+1)^2} & \frac{sq(s)}{(s+1)^2} \\ \frac{q(s)}{s+1} & \frac{-q(s)}{s+1} & \frac{q(s)}{(s+1)^2} & \frac{q(s)}{(s+1)^2} \\ 0 & 0 & \frac{s}{(s+1)^2} & \frac{-s^2}{(s+1)^2} \\ 0 & 0 & \frac{s}{(s+1)^2} & \frac{s}{(s+1)^2} \end{bmatrix}$$

Now, an inspection of this matrix will reveal that it has four independent columns over the field of complex numbers and hence if all four possible outputs are used, we will have $\delta_{\min} = 0$ implying that the fault diagnosis equations have locally unique solutions. On the other hand, if only two outputs, E_o and i_c , are measured, our modified Jacobian matrix will reduce to the first and third rows of the matrix shown in equation 31. which has column rank 3. As such, if we only use these two test outputs, we obtain $\delta_{\min} = 1$ and hence the solution to the fault diagnosis equations will be characterized by a single arbitrary parameter.

In this latter case, with only E_o and i_c taken as test outputs, theorem 2 implies that dF/dr will have rank 3 for almost any choice of $3 = n - \delta_{\min}$ test frequencies. Choosing $s_1 = 1$, $s_2 = 2$, and $s_3 = 3$, we obtain

32.

$$\frac{dF}{dr}(r_o) = \begin{bmatrix} g(1)/2 & 0 & g(1)/4 & g(1)/4 \\ 0 & 0 & 1/4 & -1/4 \\ \hline 2g(2)/3 & 0 & 2g(2)/9 & 2g(2)/9 \\ 0 & 0 & 2/9 & -2/9 \\ \hline 3g(3)/4 & 0 & 3g(3)/16 & 3g(3)/16 \\ 0 & 0 & 3/16 & -3/16 \end{bmatrix}$$

which has three linearly independent columns as long as $g(1) \neq 0$, $g(2) \neq 0$ and $g(3) \neq 0$. Indeed, in this example, any two of the three frequencies would have sufficed to yield three linearly independent columns. Note, for scalar transfer functions, theorem 2 implies that $n - \delta_{\min}$ frequencies are actually required but for matrix transfer functions fewer frequencies may suffice.

Of course, for the circuit of Figure 2, we have a choice of some 15 combinations of the four outputs with which we may choose to work for the diagnosis of the circuit. The resultant δ_{\min} 's for the various combinations of outputs are given in table 1.⁵

Finally, with the aid of Table 1, one may readily develop a test point selection algorithm for our circuit.^{4,5} For instance, if we desire to find the smallest set of outputs which yield a $\delta_{\min} \leq 1$ an inspection of the table will reveal that E_o and i_L , i_L and i_C , or E_o and i_C are the optimal choices. Of course, if one attributes a cost to the various outputs (determined by the convenience of making the required measurements), then we may further distinguish between these three possibilities. For instance, if voltage measurements are deemed to be easier than current measurements, the combination of i_L and i_C may be excluded with the decision between the remaining two options being dependent on whether it is easier to measure the circuit's input current (i_C) or its load current (i_L).

Outputs	δ_{\min}
E_o, i_L, i_C, V_i	0
E_o, i_L, i_C	0
i_L, i_C, V_i	1
i_L, V_i, E_o	1
V_i, E_o, i_C	1
E_o, i_L	1
i_L, i_C	1
i_L, V_i	2
V_i, E_o	2
E_o, i_C	1
i_C, V_i	2
E_o	2
i_L	2
i_C	2
V_i	3

Table 1: Measure of testability for the circuit of Figure 2 using various combinations of test outputs.

As a second example, consider the one stage transistor amplifier shown in Figure 3 with the AC equivalent circuit of Figure 4. Since it is clearly impossible to distinguish between failures in the two parallel bias resistors, R_a and R_b , these two resistors have been combined into the single resistor, R'_a in the component connection model of equations 33. and 34. Taking all of the component parameters as potentially faulty, r becomes a 12 vector composed of C_1, r_x, \dots, R_L and as before, we take all parameters to have the nominal value of unity.

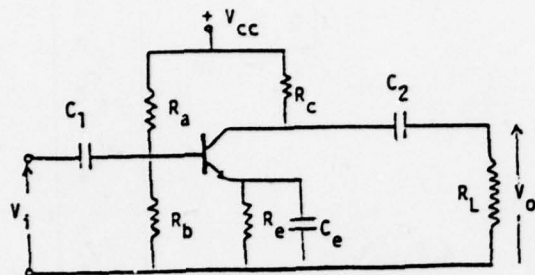


Figure 3. One Stage Transistor Amplifier

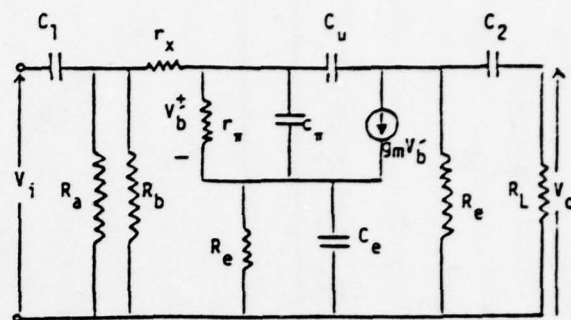


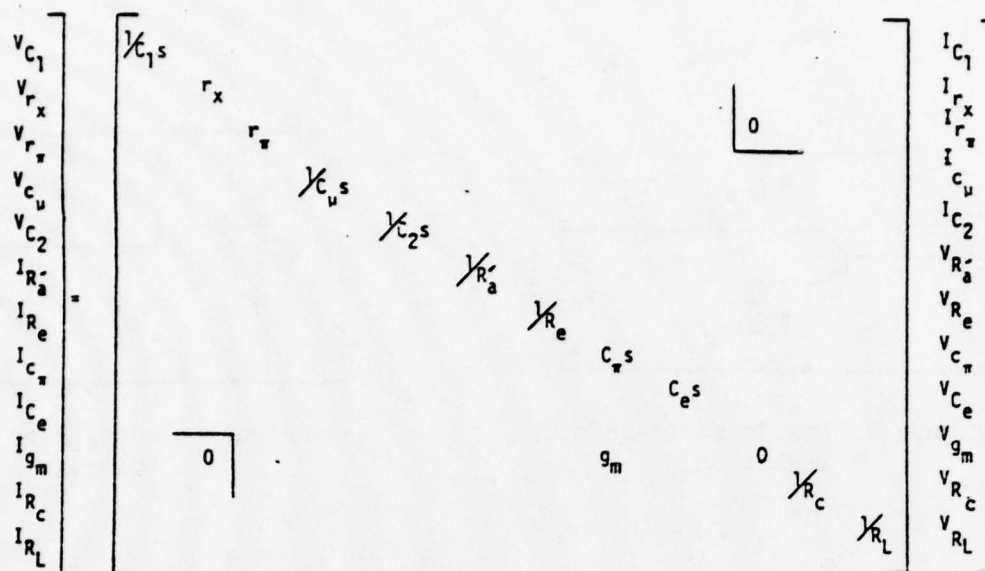
Figure 4. Amplifier Equivalent Circuit

33.

I_{C_1}	0	0	0	0	0	1	1	0	1	0	1	1	0	V_{C_1}
I_{r_x}	0	0	0	0	0	0	1	0	1	0	1	1	0	V_{r_x}
I_{r_π}	0	0	0	0	0	0	1	-1	1	-1	0	0	0	V_{r_π}
I_{C_u}	0	0	0	0	0	0	0	0	0	1	1	1	0	V_{C_u}
I_{C_2}	0	0	0	0	0	0	0	0	0	0	0	1	0	V_{C_2}
V_{R_a}	-1	0	0	0	0	0	0	0	0	0	0	0	1	I_{R_a}
V_{R_e}	-1	-1	-1	0	0	0	0	0	0	0	0	0	1	I_{R_e}
V_{C_π}	0	0	1	0	0	0	0	0	0	0	0	0	0	I_{C_π}
V_{C_e}	-1	-1	-1	0	0	0	0	0	0	0	0	0	1	I_{C_e}
V_{g_m}	0	0	1	-1	0	0	0	0	0	0	0	0	0	I_{g_m}
V_{R_c}	-1	-1	0	-1	0	0	0	0	0	0	0	0	1	I_{R_c}
V_{R_L}	-1	-1	0	-1	-1	0	0	0	0	0	0	0	1	I_{R_L}
V_o	-1	-1	0	-1	-1	0	0	0	0	0	0	0	1	V_i
I_{C_1}	0	0	0	0	0	1	1	0	1	0	1	1	0	
V_{R_a}	-1	0	0	0	0	0	0	0	0	0	0	0	1	
I_e	0	0	0	0	0	0	1	0	1	0	0	0	0	

Once again we let the input voltage be the only test input for the system and we take V_o , I_{C_1} , V_{R_a} , and I_e , to be possible output test points. The resultant δ_{\min} for each of the 15 possible combinations of these output terminals is tabulated in in Table 2.⁵

34.



Outputs	δ_{\min}
V_o	3
I_{C1}	2
$V_{R'a}$	2
I_e	3
V_o, I_{C1}	0
$V_o, V_{R'a}$	1
V_o, I_e	0
$I_{C1}, V_{R'a}$	2
I_{C1}, I_e	1
$V_{R'a}, I_e$	0
$V_o, I_{C1}, V_{R'a}$	0
V_o, I_{C1}, I_e	0
$V_o, V_{R'a}, I_e$	0
$I_{C1}, V_{R'a}, I_e$	0
$V_o, I_{C1}, V_{R'a}, I_e$	0

Table 2: Measure of Testability for the circuit of Figure 3 using various test outputs.

From the table it is apparent that no single test output suffices to yield a $\delta_{\min} = 0$ (perfect testability) though $\delta_{\min} = 0$ can be achieved using two test outputs; V_o and I_{C_1} or V_o and I_e .

VI. Conclusions

Our purpose in the preceeding has been to formulate an analytic theory in support of the intuitive art usually associated with the design of a test algorithm. With the aid of the techniques developed above, we believe that it will be possible to develop an automated test program generation (ATPG) algorithm for linear systems.^{4,5} Indeed, such an algorithm could be readily combined with the same computer-aided design (CAD) algorithm used in the system design process.⁹ Given the component connection equations such an algorithm could be employed to automatically (or interactively) choose test points and test frequencies and generate the required set of fault diagnosis equations. These could then be stored on tape and supplied to the automatic test equipment (ATE) in which a faulty system would be tested and the fault diagnosis equations solved.

Although we do not propose to discuss the actual solution of the fault diagnosis equations here, it should be pointed out that by assuming that relatively few components have failed, say $p \ll n$, it is possible to develop specialized algorithms for the solution of the fault diagnosis equations which are far more efficient than standard equation solvers in this application.^{7,11,12} These are typically derived from the fault simulation algorithms used in the diagnosis of digital systems and may naturally be classified into "simulation before test" and "simulation after test" algorithms. Some of the algorithms are discussed in references 7, 9, 10 and 11.

Finally, we note that as formulated above, the measure of testability, δ_{\min} , assumes that any combination component failures is possible. If, however, we assume that at most $p \ll n$ components fail simultaneously, the ambiguity in the

solution of the fault diagnosis equations may actually be less than δ_{\min} . For instance, in the example of Figure 3, with only V_o taken as an output $\delta_{\min} = 3$, yet the fault diagnosis equations can be solved exactly if we assume that only one parameter is out of tolerance.¹⁰ The point, here, is that even though the solution of the fault diagnosis equations in n -space has three arbitrary parameters when the solution is restricted to the one dimensional manifold of parameter vectors in which all but one coordinant are nominal it is unique.

VII. References

1. Dejka, W.J., "Measure of Testability in Device and System Design," Proc. of the 20th Midwest Symp. on Circuits and Systems, Lubbock, Texas, August, 1977, pp. 39-52.
2. Dejka, W.J., "A Review of Measurements of Testability for Analog Systems," Proc. of the 1977 AUTOTESTCON, Hyannis, Mass., November 1977, pp. 279-284.
3. Sen, N. and R. Saeks, "A Measure of Testability and its Application to Test Point Selection - Theory," Proc. of the 20th Midwest Symp. on Circuits and Systems, Lubbock, Tx., August 1977, pp. 576-583.
4. Sen, N. and R. Saeks, "A Measure of Testability and its Application to Test Point Selection - Computation," Proc. of the 1977 AUTOTESTCON, Hyannis, Mass., November 1977, pp. 212-219.
5. Sen, N., M.S. Thesis, Texas Tech Univ., Lubbock, TX, 1977.
6. Ransom, M.N., and R. Saeks, "A Functional Approach to Fault Analysis in Linear Systems," in Rational Fault Analysis, New York, Marcel Dekker, 1977, pp. 124-134.
7. Saeks, R. and R.A. DeCarlo, Interconnected Dynamical Systems, New York, Marcel Dekker, (to appear).
8. Singh, S.P. and R. -W. Liu, "Existence of State Equation Representation of Linear Large-Scale Dynamical Systems," IEEE Trans. on Circuits and Systems, Vol. CAS-20, pp. 239-246, (1973).
9. Proc. of the Workshop on Automatic Test Technology, NSIA, San Diego, April, 1978, (to appear).

10. Chen, H.M.S., and R. Saeks, "A Search Algorithm for the Solution of the Fault Diagnosis Equations," Unpublished Notes, Texas Tech Univ., 1978.
11. Chen, H.M.S., M.S. Thesis, Texas Tech Univ., 1977.
12. Ransom, M.N. and R. Saeks, "The Connection Function - Theory and Application," Int. J. on Circuit Theory and its Applications, Vol. 3, pp. 5-21, (1975).
13. Fleming, W., Functions of Several Variables, Addison- Wesley, Reading, 1965.
14. Spivak, M., Calculus on Manifolds, Benjamin, Amsterdam, 1966.

A SEARCH ALGORITHM FOR THE SOLUTION OF THE
FAULT DIAGNOSIS EQUATIONS*

H.M.S. Chen** and R. Saeks

*This research supported in part by Office of Naval Research Grants 75-C-0924 and 76-C-1136.

**Presently with the National Semiconductor Corporation, San Jose, Calif.

Abstract

A search algorithm for the solution of the fault diagnosis equations arising in linear time invariant analog circuits and systems is presented. By exploitation of Householder's formula an efficient algorithm whose computational complexity is a function of the number of system failures rather than the number of system components is obtained.

Introduction

Conceptually, the fault analysis problem for an analog circuit or system amounts to the measurement of a set of externally accessible parameters of the system from which one desires to determine the internal system parameters or equivalently¹ locate the failed components as illustrated in Figure 1. Here, the

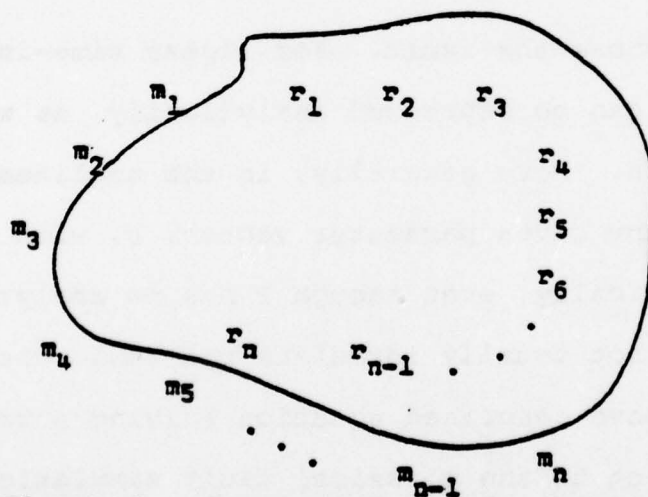


Figure 1. Conceptual Model of Fault Diagnosis Problem.

measurements, m_1 , may represent data taken at distinct test points or alternatively, data taken at a fixed test point under different stimuli.

¹Since the problem of determining the values of the failed components is usually straightforward, once the failures have been located, the exact determination of all internal component parameters is essentially equivalent to the problem of "simply" locating the failed components.

Similarly, the r_i represent parameters characterizing the various internal system components. Here, a single parameter may characterize an entire component, say a resistance, capacitance or inductance. Alternatively, a component may be represented by several parameters: the h-parameters of a transistor, the poles and gain of an op-amp, etc. In general, one models a system component by the minimum number of parameters which will allow the failure to be isolated up to a "shop replaceable assembly" with all "allowed" system failures manifesting themselves in the form of some parameter change.

To solve the fault diagnosis problem, one then measures $m = \text{col}(m_i)$ and solves a nonlinear algebraic equation

$$1. \quad m = F(r)$$

for $r = \text{col}(r_i)$ to diagnose the fault. For linear time-invariant systems the function F can be expressed analytically, as we shall see in the following section. More generally, in the nonlinear case, one can evaluate $F(r)$ for any given parameter vector, r , with a simulator, and thus solve 1. numerically, even though F has no analytic expression.

Although one does not usually formulate the fault diagnosis problem in terms of the above described equation solving notation, this formulation is equivalent to the classical fault simulation concept.⁹ Indeed, fault simulation is simply a search algorithm for solving 1. Here, one precomputes $\hat{m} = f(\hat{r})$ for each allowable¹ faulty parameter vector \hat{r} and then compares the measured m with the simulated \hat{m} 's, stored in a fault dictionary, to solve equation 1.

¹ By allowable faults we mean all possible parameter vectors, \hat{r} , which satisfy a specified set of fault hypotheses. These typically restrict the maximum number of component parameters which are simultaneously out of tolerance and the type of failure (open circuit, short circuit, small change, etc.)

Although the above described approach to fault simulation has been successful¹ when applied to digital system, there is considerable question surrounding its applicability to analog circuits and systems¹. The problem here is two-fold. First, rather than simply failing as a one or zero, an analog parameter has a continuum of possible failures. Secondly, unlike a digital system wherein a component is either good or bad, in an analog system, a component parameter is either in tolerance or out of tolerance. As such, for each hypothesized failure, it may prove necessary to do an entire family of Monte Carlo simulations in which the values of the good components are randomly chosen within their tolerance limits. Although, at the present time we have insufficient practical experience to determine the precise number of fault simulations required for analog fault diagnosis, it is estimated that the number of simulations required for an analog system will exceed the number of simulations required for a digital system of similar complexity by a factor ranging between two and six order of magnitude.¹ As such, the fault simulation concept which has proven to be so successful for a digital system may not be applicable in the analog case.

As an alternative to fault simulation, one may adopt one of the more classical equation solving algorithms for the solution of 1.^{2,3} Here, one first measures m and on the basis of this measurement, makes an initial guess r^0 (usually taken to be nominal parameter vector) at the solution of the equations. One then evaluates $m^0 = F(r^0)$ and compares it with m . If $m^0 = m$, r^0 is the solution to the fault diag-

¹ Most industrial users of ATE obtain satisfactory fault detection in digital circuits via fault simulation techniques but require guided probe techniques in addition to the fault dictionary data for fault diagnosis (isolation).

nosis equation. If not, one makes a new "educated" guess at the solution, r^1 , (usually based on the deviation between m and m^0) and repeats the process by evaluating $m^1 = F(r^1)$ and comparing it with m . Hopefully, sequence of component parameter vectors, r^i , and simulated data vectors, $m^i = F(r^i)$, is obtained which "quickly" converges to r and m , respectively. Since the evaluation of $F(r^i)$ is essentially equivalent to the simulation of the system with the faulty parameter values, r^i , this technique is really another form of fault simulation. In this case, however, one simulates the system after the data vector has been measured and uses this data to make an educated guess at a (hopefully) small number of parameter vectors at which the system should be simulated. As such, the approach has been termed simulation after test¹ to distinguish it from the classical approach wherein all simulation is done before test.¹

At the time of this writing, both approaches are under study¹, neither of which have been shown to be superior. Fault "simulation after test" requires that one include an efficient simulator in the ATE itself, which can be used for on-line computation of $m^i = F(r_i)$ after the UUT has been measured. On the other hand, simulation after test eliminates the requirement of searching a large fault dictionary for the (approximate) data matches required by "simulation before test." In addition, the complex ATPG requirement for "simulation before test" is eliminated.

To make "simulation after test" feasible, however, an efficient equation solving algorithm is required to obtain convergence of the r^i sequence in a reasonable amount of time. Moreover, since "real world" failures in analog circuits and systems often take the form of open and

short circuited components or large parameter deviations from nominal the classical perturbational algorithms a-la Newton-Raphson are inapplicable. Fortunately, in the context of the fault diagnosis problem, one can reasonably assume that relatively few component parameters have failed. As such, even though it is not valid to assume that $r-r^0$ (the deviation of r from nominal) is small in norm, it is reasonable to assume that it is small in "rank." The purpose of the present paper is to formulate a search algorithm for the solution of the fault diagnosis equations which exploits such an assumption.

In the following section, the explicit form for the fault diagnosis equations arising in linear time-invariant circuits and systems is derived.³ Householder's formula⁴ is then used to exploit the special form of these equations in combination with an assumption that r differs from r^0 in relatively few coordinates to formulate a search algorithm for the solution of the fault diagnosis equations in which the computational complexity of the simulation process is a function of the number of the failures rather than the number of components. This algorithm is based on a similar algorithm suggested by Temes⁵ for "simulation before test" and a large-change sensitivity algorithm first given by Leung and Spence.⁶ Finally, examples of the application of the algorithm to active and passive circuits are presented and a study of the robustness of the algorithm to deviations of the "good" components from their nominal values is presented.⁷

Explicit Form of the Fault Diagnosis Equations

In the case of a linear time-invariant circuit or system, the fault diagnosis equations discussed abstractly in the previous section, may be expressed explicitly in analytical form. Indeed, it is the ex-

explicit nature of this form which makes our simplified solution algorithm possible. Since the fault diagnosis equations deal with the relationship between the externally measureable system parameters, m , and the internal component parameters, r , we adopt a "component connection model" as the starting point for the derivation of the fault diagnosis equations.⁸ This is one of several commonly employed large scale system models in which the components and connections in a circuit or system are modeled by distinct equations, thereby permitting one to explicitly deal with the relationship between the individual component parameters and the composite system parameters.

Since the present study is restricted to linear time-invariant systems, we assume that each component is characterized by a transfer function matrix which is dependent on the potentially variable component parameters, $Z_i(s, r)$. For the classical RLC components $Z_i(s, r)$ may take the form R , Ls , or $1/sC$ for the case of a resistor, inductor, or capacitor, respectively. More generally, one may model an op-amp by the transfer function $k/(s-p_1)(s-p_2)$ where the parameter vector, r , now represents the three potentially variable component parameters; k , p_1 , and p_2 ; or a delay by ke^{sT} , etc. Although the symbol Z is used (for historical reasons), the components are not assumed to be represented by impedance matrices. Indeed, hybrid models are used in most of our examples. For the purpose of analysis, it is assumed that all faults manifest themselves in the form of changes, possibly catastrophic, in the parameter vector, r , with the frequency characteristics of the components unchanged. Although not universal, this fault hypothesis covers the most commonly encountered situations and subsumes the common industrial practice of assuming that all failures in analog circuits

and systems take the form of open and short circuited components.¹

Our system components are thus characterized by a set of simultaneous equations

$$2. \quad b_i = Z_i(s,r)a_i \quad i = 1, 2, \dots, n$$

where a_i and b_i denote the component input and output vectors, respectively. For notational brevity, these component equations may be combined into a single block diagonal matrix equation

$$3. \quad b = Z(s,r)a$$

where $b = \text{col}(b_i)$, $a = \text{col}(a_i)$ and $Z(s,r) = \text{diag}(Z_i(s,r))$. Although 3. is written as a single equation, it is important to remember that it represents a set of decoupled, simultaneous equations, in which $Z(s,r)$ is block diagonal. Indeed, we will exploit this fact in the application of Householder's formula.

Although there are many ways to represent the connection in a circuit or system; say a block diagram, linear graph or signal flow graph, any such representation is simply a graphical means for displaying a set of connection equations: Kirchoff laws, adder equations, etc. As such, for our component connection model we adopt⁸ a purely algebraic connection model in which the connection equations are displayed explicitly without the intermediary of some kind of graphical connection diagram. This takes the form

$$4. \quad a = L_{11}b + L_{12}u$$

$$y = L_{21}b + L_{22}u$$

where u and y represent the vectors of accessible inputs and outputs

which are available to the test system. In simple systems, the connection matrices, L_{ij} , are usually obtainable by inspection, whereas, in more complex systems, computer codes have been developed for their derivation.¹³ Moreover, they are assured to exist in all but the most pathological systems.⁸

It is the pair of simultaneous matrix equations 3 and 4 which are termed the component connection model. By combining equations 3 and 4 to eliminate the component input and output variables, a and b , one may derive^{3,8} an expression for the transfer function matrix observable by the test system between the test input and output vectors, u and y , obtaining

$$5. \quad S(s,r) = L_{22} + L_{21}(1 - Z(s,r)L_{11})^{-1}Z(s,r)L_{12}$$

where

$$6. \quad y = S(s,r)u$$

For a linear time-invariant system the transfer function $S(s,r)$ is a complete description of the measurable data about the UUT available to the test system. Moreover, being rational it is completely determined by its value at a finite number of frequencies. As such, without loss of generality we may take our vector of measured data to be of the form

$$7. \quad m = \text{col}[S(s_1,r), S(s_2,r), \dots, S(s_k,r)]$$

The fault diagnosis equations then take the form

$$10. \quad A^{-1} = [1 - B^{-1}C(1 + DB^{-1}C)^{-1}D]B^{-1}$$

As such, once B^{-1} is known, one may compute the inverse of the nxn matrix, A, in terms of B^{-1} and the inverse of the pxp matrix $(1 + DB^{-1}C)$. This technique has been used effectively for large change sensitivity analysis⁶ and has recently been suggested by Temes for application to fault simulation.⁵ This is achieved by exploiting the block diagonal character of $Z(s,r)$. Thus if r differs from r^0 in q coordinates $Z(s,r)$ will differ from $Z(s,r^0)$ only in the pxp block composed of components which are effected by the faulty parameters¹. If the rows and columns of $Z(s,r)$ are re-ordered so that this block appears in the upper left corner of $Z(s,r)$ then,

$$11. \quad Z(s_i, r) = Z(s_i, r^0) + \begin{bmatrix} \Delta & | & 0 \\ \hline 0 & | & 0 \end{bmatrix}$$

where Δ is pxp and $Z(s,r)$ is nxn. We then have

$$12. \quad (1 - Z(s_i, r)L_{11}) = (1 - Z(s_i, r^0)L_{11}) + \begin{bmatrix} -\Delta(s_i, r) \\ \hline 0 \end{bmatrix} L_{11}^p$$

where L_{11}^p denotes the upper (after reordering) p rows of L_{11} .

Finally, an application of Householder's formula yields

¹ Here, p is the sum of the dimensions of all the blocks of $Z(s,r)$ which are dependent on the q coordinates in which r differs from r^0 . Typically, $q = p$ with the exact relationship depending the block sizes.

$$\begin{aligned}
 13. \quad & (1 - Z(s_i, r) L_{11})^{-1} \\
 & = \left[1 - (1 - Z(s_i, r^0) L_{11})^{-1} \begin{bmatrix} -\Delta(s_i, r) \\ \hline 0 \end{bmatrix} \right. \\
 & \quad \left. \left(1 + L_{11}^p (1 - Z(s_i, r^0) L_{11})^{-1} \begin{bmatrix} -\Delta(s_i, r) \\ \hline 0 \end{bmatrix} \right)^{-1} L_{11}^p \right] (1 - Z(s_i, r^0) L_{11})^{-1}
 \end{aligned}$$

Although quite complex, the only major matrix computation required for the inversion of $(1 - Z(s_i, r) L_{11})$ via 13 is the inversion of the $p \times p$ matrix in parentheses. As such, as long as the number of faulty parameter values remains small, equation 13 represents an extremely efficient means of carrying out a large number of fault simulations with relatively little computational capacity. Although Temes originally suggested the technique in the context of a "simulation before test" algorithm, the above application of Householder's formula is ideally suited for "simulation after test", wherein, it reduces the computational requirements for the simulation process to well within the capabilities of the minicomputers usually found in modern ATE.

Although Householder's formula yields an efficient means for solving the fault diagnosis equations once the faulty parameters have been determined, it remains to locate the set of faulty parameters. Fortunately, the efficiency of the solution algorithm based on Householder's formula is such that one can justify a search through "all" allowable sets of faulty parameters to locate the actual failures. Indeed, if we denote the "reduced fault diagnosis equations" in which all component values are assumed to be nominal except for q specified parameters; $r_{i(1)}, r_{i(2)}, \dots, r_{i(q)}$; by

$F_{i(1)}, i(2), \dots, i(q)$ then the equation

$$14. \quad m = F_{i(1), i(2), \dots, i(q)}(r_{i(1)}, r_{i(2)}, \dots, r_{i(q)})$$

will have a solution if and only if the faulty parameter values are among the $r_{i(1)}, r_{i(2)}, \dots, r_{i(q)}$. As such, if one attempts to solve 14 for each allowable family of faulty parameters, the actual fault will be indicated by the existence of a solution to the equation.

Although such a search algorithm might at first seem to be highly inefficient, when one observes that with the aide of Householder's formula, the evaluation of $F_{i(1), i(2), \dots, i(q)}$ requires only the inversion of $p \times p$ ($p = q$) matrix it is seen that this is not the case. Moreover, if one searches for the most likely failures first, relatively few equations need be solved in practice. In actual implementation in a "simulation after test" algorithm, one can readily search through all possible combinations of one, two, or three simultaneous failures, and commonly encountered combinations of larger numbers of failures, thus locating the far majority of failures in a reasonable amount of ATE time.

An alternative formulation of the search algorithm which alleviates the numerical difficulties associated with the attempt to solve a set of equations which may not have a solution (as is the case whenever one attempts to solve 14 with the wrong choice of faulty parameters) is to employ an optimization algorithm, rather than an equations solver, to minimize

$$15. \quad J_{i(1), i(2), \dots, i(q)}(r_{i(1)}, r_{i(2)}, \dots, r_{i(q)})$$

$$= || m - F_{i(1), i(2), \dots, i(q)}(r_{i(1)}, r_{i(2)}, \dots, r_{i(q)}) ||^2$$

Since 15. has a zero minimum if and only if 14. has a solution a search through the minimization of 15. for all allowable sets of faulty parameters will also locate the faulty parameters (indicated by a zero minimum).

Examples

As a first example, consider the LC filter shown in Figure 2. for which

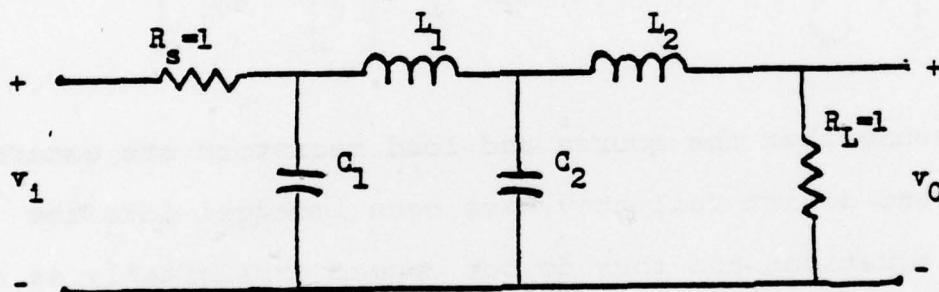


Figure 2. LC Filter.

the component connection model takes the form

$$16. \begin{bmatrix} v_{c1} \\ i_{L1} \\ v_{c2} \\ i_{L2} \end{bmatrix} = \begin{bmatrix} 1/s_{c1} & 0 & 0 & 0 \\ 0 & 1/s_{L1} & 0 & 0 \\ 0 & 0 & 1/s_{c2} & 0 \\ 0 & 0 & 0 & 1/s_{L2} \end{bmatrix} \begin{bmatrix} i_{c1} \\ v_{L1} \\ i_{c2} \\ v_{L2} \end{bmatrix}$$

and

$$17. \begin{bmatrix} i_{c1} \\ v_{L1} \\ i_{c2} \\ v_{L2} \\ \hline v_o \end{bmatrix} = \begin{bmatrix} -1 & -1 & 0 & 0 & | & 1 \\ 1 & 0 & -1 & 0 & | & 0 \\ 0 & 1 & 0 & -1 & | & 0 \\ 0 & 0 & 1 & -1 & | & 0 \\ \hline 0 & 0 & 0 & 1 & | & 0 \end{bmatrix} \begin{bmatrix} v_{c1} \\ i_{L1} \\ v_{c2} \\ i_{L2} \\ \hline v_i \end{bmatrix}$$

Since we assume that the source and load resistors are external to the filter and do not fail they have been imbedded into the connection equations and thus do not appear explicitly as components. The filter components are assumed to have the nominal values

$$18. \quad C_1 = 10, L_1 = 20, C_2 = 30, \text{ and } L_2 = 40$$

and it is assumed that no more than one component fails at a time (though the failure may be catastrophic). Our "simulation after test" fault diagnosis algorithm then requires that we minimize $J_1(C_1)$, $J_2(L_1)$, $J_3(C_2)$, and $J_4(L_2)$. The performance measure with zero minimum then represents the failed component with the minimizing value for that performance measure representing the value of the failed component. All other component values must then be nominal (since it is assumed that only one component

fails). Note: the minimizing value for the non-zero J_i 's does not correspond with the correct component values for those components.

This filter was simulated with each of its four components out of tolerance (by as much as 100 percent) with the search algorithm being applied to the simulated data. Since only one parameter is assumed to fail at a time and $Z(s,r)$ is diagonal each of the four required minimizations was carried out by purely scalar operations using a Golden Section search. In all four cases the fault was correctly located with the faulty parameter value being determined "exactly." The resultant data is summarized in Table 1. Note: in each case the minimum value for J_i for the faulty component is at least three orders of magnitude lower than the minimum value J_i for any non-faulty component. As such, the failure is easily located and one can expect the algorithm to remain viable in the face of numerical and or approximation error.

As a more sophisticated example, consider the one stage transistor amplifier of Figure 3 and its wide band equivalent circuit shown in Figure 4. Note that the parallel resistors, R_a and R_b , appearing in this model have been lumped together into a single resistance, R_s , since it is clearly impossible to distinguish between failures in these two components from external measurements.

18.

$$\begin{bmatrix} V_{c1} \\ V_{rx} \\ V_{r\pi} \\ V_{cu} \\ V_{c2} \\ I_{RS} \\ I_{Re} \\ I_{c\pi} \\ I_{Ce} \\ I_{gm} \\ I_{Rc} \\ I_{RL} \end{bmatrix} = \begin{bmatrix} 1/c_{1s} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & r_x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & r_{\pi} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/c_{us} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/C_{2s} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/R_s & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/R_e & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_{\pi s} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_{es} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & g_m & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/R_c & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/R_L \end{bmatrix} \begin{bmatrix} I_{c1} \\ I_{rx} \\ I_{r\pi} \\ I_{cu} \\ I_{c2} \\ V_{RS} \\ V_{Re} \\ V_{c\pi} \\ V_{Ce} \\ V_{gm} \\ V_{Rc} \\ V_{RL} \end{bmatrix}$$

and

19.

$$\begin{bmatrix} I_{c1} \\ I_{rx} \\ I_{r\pi} \\ I_{cu} \\ I_{c2} \\ V_{RS} \\ V_{Rc} \\ V_{c\pi} \\ V_{Ce} \\ V_{gm} \\ V_{Rc} \\ V_{RL} \\ \hline V_o \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -1 & -1 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline -1 & -1 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} V_{c1} \\ V_{rx} \\ V_{r\pi} \\ V_{cu} \\ V_{c2} \\ I_{RS} \\ I_{Rc} \\ I_{c\pi} \\ I_{Ce} \\ I_{gm} \\ I_{Rc} \\ I_{RL} \\ \hline V_i \end{bmatrix}$$

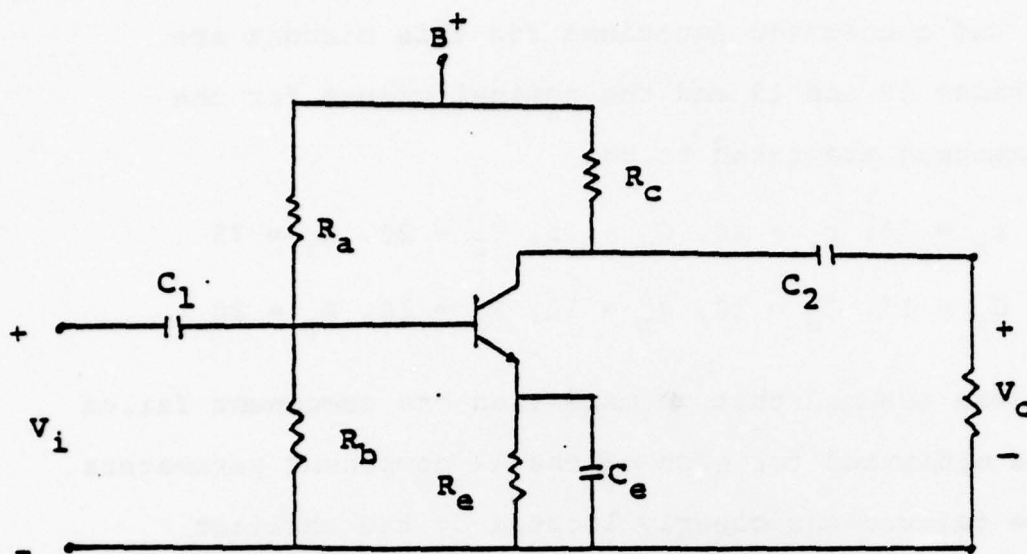


Figure 3. One stage transistor amplifier.

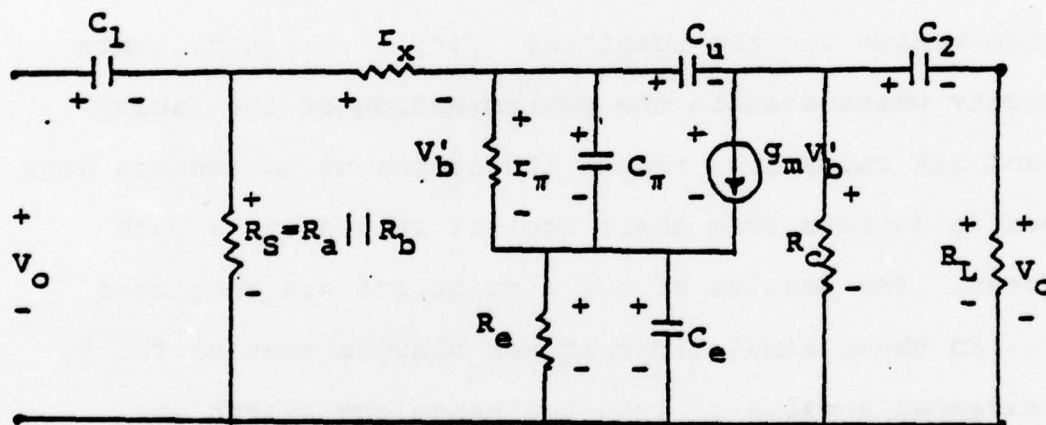


Figure 4. Amplifier equivalent circuit

The component and connection equations for this circuit are given by equations 18 and 19 and the nominal values for the component parameters are taken to be

$$\begin{aligned} C_1 &= 20, r_x = 10, r_\pi = 40, C_u = 25, C_2 = 20, R_s = 75 \\ 20. \quad R_e &= 30, C_\pi = 15, C_e = 10, g_m = 10, R_c = 10, R_l = 20 \end{aligned}$$

As before, it was assumed that no more than one component failed and $J_i(r_i)$ was minimized for each of the 12 component parameters. Once again the failure was clearly located by the smallest minima with accurate determination of the faulty parameter value. Indeed, in each case, the minimum value of $J_i(r_i)$ for the faulty parameter value is at least 5 orders of magnitude less than the minima for the remaining $J_i(r_i)$. As such, there is no ambiguity whatsoever in the determination of the faulty component and its value even though the component parameters have been allowed to deviate from their nominal values by as much as 500 percent. The results of our simulations are tabulated in Table 2. In these simulations it was assumed that no faulty parameter exceeded a value of 1000 and hence the search was stopped if the minimizing value for $J_i(r_i)$ reached 1000. This was necessitated by the requirement that the Golden Section search be restricted to a finite interval. Of course, the minimization algorithm can be easily modified to take into account infinite values of r_i ; i.e. open or short circuited components.

Multiple Failures

Although we have not given any numerical examples of the

application of the search algorithm to the case where multiple failures are assumed, the basic concept of our algorithm remains valid. Computationally, the simple one-dimensional Golden Section search used to minimize $J_i(r_i)$, however, must be replaced by a multidimensional optimization algorithm; say steepest decent, conjugate gradient, etc; to minimize $J_{i(1), i(2), \dots, i(q)}$ ($r_{i(1)}, r_{i(2)}, \dots, r_{i(q)}$) for each set of q allowable failures. In addition, each evaluation of $F_{i(1), i(2), \dots, i(q)}$ requires the inversion of a $p \times p$ matrix ($p = q$) rather than the simple scalar operations required in the single fault case.

An alternative approach to the minimization of $J_{i(1), i(2), \dots, i(q)}$ which requires only scalar mathematics is to use a cyclical one-dimensional search algorithm on the q parameters $r_{i(1)}, r_{i(2)}, \dots, r_{i(q)}$. Here, one minimizes $J_{i(1), i(2), \dots, i(q)}$ as a function of one parameter at a time cycling through the q parameters until a minimum is achieved. Although such an optimization algorithm usually requires more iterations than a true multidimensional optimization, the entire process can be carried out with scalar operations. In particular, since one only varies a single parameter at a time, Householder's formula allows $F_{i(1), i(2), \dots, i(q)}$ to be evaluated at each iteration without matrix inversion while a simple Golden Section search may be used for each one-dimensional minimization.

Robustness

Unlike the case of fault diagnosis in a digital system wherein

a component is unambiguously good or bad, in an analog circuit or system, a component parameter is either in tolerance or out of tolerance. As such, any fault diagnosis algorithm which makes use of the nominal component parameters must be tested for robustness. I.e. how effective is the algorithm at locating the faulty component(s) when the good components are not precisely equal to their nominal values. As such, our search algorithm for fault diagnosis was applied to the transistor amplifier using simulated measurements in which one component was out of tolerance (taken to be 10 percent) and the remaining component parameters were in tolerance but not equal to their nominal values.⁷ Of course, the nominal values are used to define the F_1 since the actual value of the good components is unknown. Not surprisingly, this results in some ambiguity in the diagnosis process since $J_i(r_i)$ can never be reduced exactly to zero. Fortunately, the data of Table 2 resulting from our "perfect" simulation, indicates that we have five orders of magnitude in which to work. As such, our simulation yielded good though not perfect results. In particular, the algorithm correctly located the fault in 71 percent of the trials with an ambiguity group of one in 50 percent of these cases and ambiguity groups of two, three and four in the remaining cases. Here, the ambiguity group was taken to be the set of all r_i for which the minimum value of J_i was of the smallest order of magnitude achieved by any of the performance measures. The results of some typical simulations are shown in Table 3. Since all of the good components in this simulation were taken to be at the limits of their tolerance interval, these results actually

represent a worst case situation. As such, we believe that the search algorithm will yield significantly better results in a "real world" situation, wherein most of the components will have near nominal values with relatively few of the "good" component parameters lying near their tolerance limits.

Hybrid Algorithms

Although the terminology has only recently been formulated¹, most of the algorithms which have been proposed over the years for the solution of the fault analysis problem in analog circuits and systems can naturally be categorized as either "simulation before test" or "simulation after test" algorithms.⁹ Although the preceding development has been presented in the context of a "simulation after test" algorithm, many of the techniques, such as the application of Householder's formula⁵, are also applicable to "simulation before test" algorithms. Indeed, the techniques are ideally suited to a hybrid algorithm. Here, one would employ a two-pass diagnostic algorithm wherein the measured data vector, m , is first compared with pre-simulated data stored in a fault dictionary. If the fault is so located, the diagnosis process is terminated. If the fault is not located among those which have been presimulated and stored in the fault dictionary, the hybrid algorithm will then revert to a "simulation after test" mode until a sequence of parameter vectors, r_1 , and simulated data vectors, m_1 , have been computed which converge to the solution of the fault diagnosis equations. At the same time the results of each of these "after test" simulations are stored in the fault dictionary

for use in future applications of the test algorithm. As such, a fault dictionary is slowly built up which includes simulations of those failures which are most commonly encountered in actual practice. Such, a hybrid algorithm would seem to achieve the best of both worlds. Common faults would be found quickly on the first pass, yet the system would still have the "simulation after test" algorithm upon which to fall back when encountering a new failure mode. Moreover, ATPG requirements would be greatly reduced with only the most common faults (say open and short circuits, single failures, etc.), being pre-simulated and the remainder of the fault dictionary being adaptively generated by the "simulation after test" algorithm as new fault modes are encountered. Such a hybrid scheme alleviates the necessity of determining the fault modes of a system in advance, as required for "simulation before test" while simultaneously eliminating the duplicate simulations of common faults required for "simulation after test".

Conclusions

Our purpose in the preceding has been the formulation of a class of techniques which we believe can serve as the basis of an effective algorithm for fault diagnosis in linear analog circuits and systems. These techniques have proven to be effective in the situation where all good component parameters are "near" nominal and give promise of sufficient robustness to cope with the "real world" situation, in which the good component parameters are in tolerance though not nominal.

Although the presentation has been formulated in the context of a "simulation after test" algorithm, the techniques presented are also applicable to "simulation before test" and hybrid algorithms.

References

1. "Report of the Industry-Joint Services Automatic Test Task Force", San Diego, April, 1977, (to appear).
2. Saeks, R., Singh, S.P., and R.W. Liu, "Fault Isolation via Components Simulation", IEEE Trans. on Circuit Theory, Vol. CT-19, pp. 634-640, (1972).
3. Ransom, M.N., and R. Saeks, "A Functional Approach to Fault Analysis in Linear Systems", in Rational Fault Analysis, (eds. R. Saeks and S.R. Liberty), New York, Marcel Dekker Inc., 1977, pp. 124-134.
4. Householder, A.S., "A Survey of Some Closed Methods for Inverting Matrices", SIAM Jour. on Appl. Math., Vol. 5, pp. 155-169, (1957).
5. Temes, G.C., "Efficient Methods for Fault Simulation", Proc. of the 20th Midwest Symp. on Circuits and Systems, Texas Tech Univ., Aug. 1977, pp. 191-194.
6. Leung, K.H., and R. Spence, "Multiparameter Large-Change Sensitivity Analysis and Systematic Exploration", IEEE Trans. on Circuits and Systems, Vol. CAS-22, pp. 796-804, (1975).
7. Chen, H.S.M., M.S. Thesis, Texas Tech University, Lubbock, Texas, 1977.
8. Saeks, R., and R.S. DeCarlo, Interconnected Dynamical Systems, New York, Marcel Dekker Inc., 1979 (to appear).
9. Saeks, R., and S.R. Liberty, Rational Fault Analysis, New York, Marcel Dekker Inc., 1977.
10. Sen, N., M.S. Thesis, Texas Tech University, Lubbock, Tx., 1977.
11. Sen, N., and R. Saeks, "A Measure of Testability and its Application to Test Point Selection - Computation", Proc. of AUTOTESTCON '77, pp. 212-219, Hyannis, Mass., Nov. 1977.
12. Sen, N., and R. Saeks, "A Measure of Testability and its Application to Test Point Selection - Theory", Proc. of the 20th Midwest Symp. on Circuits and Systems, Texas Tech Univ., Lubbock, Texas, August 1977, pp. 576-583.
13. Trauboth, H., and N. Prasad, "MARSYAS - A Software Package for Digital Simulation of Physical Systems", Proc. of the Spg. Joint Comp. Conf., pp. 223-235, 1970.

Component	C_1	L_1	C_2	L_2
Actual Parameter Values	20	20	30	40
Minimum Value for J_i	1.25×10^{-12}	5.88×10^{-8}	8.46×10^{-9}	1.96×10^{-6}
Minimizing Component Value	20	30.66	39.87	51.89

Component	C_1	L_1	C_2	L_2
Actual Parameter Values	10	40	30	40
Minimum Value for J_i	1.62×10^{-7}	2.10×10^{-4}	2.42×10^{-7}	6.94×10^{-6}
Minimizing Component Value	28.75	40	48.5	62.2

Component	C_1	L_1	C_2	L_2
Actual Parameter Value	10	20	50	40
Minimum Value for J_i	2.91×10^{-8}	2.73×10^{-7}	1.47×10^{-13}	6.87×10^{-6}
Minimizing Component Value	30.27	41.62	50	64.01

Component	C_1	L_1	C_2	L_2
Actual Parameter Value	10	20	30	45
Minimum Value for J_i	3.93×10^{-7}	4.93×10^{-7}	4.13×10^{-7}	1.04×10^{-13}
Minimizing Component Value	14.18	24.46	34.13	45

Table 1. Fault Analysis for LC Filter

Component	C_1	r_x	r_π	C_u	C_2	R_s	R_e	C_π	C_e	g_m	R_C	R_L
Actual Parameter Value	50	10	40	25	20	75	30	15	10	10	10	20
Minimum Value for J_i	6.2×10^{-14}	2.6×10^{-4}	3.1×10^{-5}	3.0×10^{-5}	1.58×10^{-5}	4.5×10^{-5}	5.3×10^{-4}	3.0×10^{-5}	1.7×10^{-4}	3.4×10^{-4}	1.2×10^{-5}	1.8×10^{-6}
Minimizing Component Value	50	88.5	0.44	28.77	1000	1000	62.6	80.1	9.2	4.5	18.2	41.9

Component	C_1	r_x	r_π	C_u	C_2	R_s	R_e	C_π	C_e	g_m	R_C	R_L
Actual Parameter Value	20	30	40	25	20	75	30	15	10	10	10	20
Minimum Value for J_i	1.1×10^{-3}	3.2×10^{-12}	2.1×10^{-2}	2.1×10^{-3}	1.3×10^{-3}	1.3×10^{-3}	8.5×10^{-3}	1.9×10^{-3}	3.5×10^{-2}	1.6×10^{-4}	7.4×10^{-2}	9.0×10^{-4}
Minimizing Component Value	1.68	30	1000	14.6	0.9	0.96	7.56	0.1	16.9	1000	1.2	2.0

Component	C_1	r_x	r_π	C_u	C_2	R_s	R_e	C_π	C_e	g_m	R_C	R_L
Actual Parameter Value	20	10	90	25	20	75	30	15	10	10	10	20
Minimum Value for J_i	1.1×10^{-9}	7.8×10^{-9}	7.6×10^{-20}	3.3×10^{-11}	2.1×10^{-9}	2.2×10^{-9}	5.2×10^{-9}	1.2×10^{-8}	4.1×10^{-8}	1.3×10^{-9}	6.2×10^{-10}	6.9×10^{-10}
Minimizing Component Value	19.9	10.01	90	24.97	19.8	72.7	29.9	14.6	10.01	10.09	9.96	19.9

Table 2. Fault Analysis for Transistor Amplifier

Component	C_1	r_x	r_w	C_u	C_2	R_s	R_e	C_w	C_e	g_m	R_C	R_L
Actual Parameter Value	20	10	40	60	20	75	30	15	10	10	10	20
Minimum Value for J_1	5.6×10^{-3}	3.4×10^{-3}	3.0×10^{-3}	8.1×10^{-3}	9.5×10^{-3}	1.4×10^{-2}	6.0×10^{-3}	1.7×10^{-3}	2.0×10^{-3}	3.1×10^{-3}	2.7×10^{-3}	4.2×10^{-3}
Minimizing Component Value	1000	5.79	0.1	60	1000	1000	1000	1000	6.65	0.72	1000	1000

Component	C_1	r_x	r_w	C_u	C_2	R_s	R_e	C_w	C_e	g_m	R_C	R_L
Actual Parameter Value	20	10	40	25	50	75	30	15	10	10	10	20
Minimum Value for J_1	4×10^{-9}	9.7×10^{-5}	1.6×10^{-5}	1.7×10^{-5}	1.7×10^{-5}	8×10^{-5}	6.3×10^{-5}	1.2×10^{-6}	6.8×10^{-5}	1.2×10^{-5}	1.4×10^{-5}	6.9×10^{-6}
Minimizing Component Value	30.9	9.4	0.88	26.9	50	1000	43	43	9.6	6.3	13.5	28.7

Component	C_1	r_x	r_w	C_u	C_2	R_s	R_e	C_w	C_e	g_m	R_C	R_L
Actual Parameter Value	20	10	40	25	20	15	30	15	10	10	10	20
Minimum Value for J_1	5.2×10^{-6}	1.6×10^{-6}	4.7×10^{-4}	2.1×10^{-4}	4.1×10^{-4}	9.1×10^{-8}	3.2×10^{-5}	3.1×10^{-4}	1×10^{-4}	2.4×10^{-4}	1.8×10^{-5}	9.4×10^{-6}
Minimizing Component Value	12.3	11.1	1000	22.6	9.5	15	20.5	0.1	10.8	50.9	6.9	13

Table 2. Fault Analysis for Transistor Amplifier (Cont:)

Component	C_1	r_x	r_n	C_u	C_2	R_g	R_e	C_n	C_e	q_m	R_c	R_L
Actual Parameter Value	20	10	40	25	20	75	80	15	10	10	10	20
Minimum Value for J_i	6.8×10^5	5.5×10^4	1.5×10^4	1.5×10^4	4.3×10^4	6.6×10^5	2.2×10^4	5.9×10^{-14}	4.2×10^4	6.4×10^4	1.4×10^4	8.8×10^5
Minimizing Component Value	61.4	8.9	0.4	29.2	1000	1000	80	78.3	9.2	4.6	19.6	47.8

Component	C_1	r_x	r_n	C_u	C_2	R_g	R_e	C_n	C_e	q_m	R_c	R_L
Actual Parameter Value	20	10	40	25	20	75	30	55	10	10	10	20
Minimum Value for J_i	8.5×10^5	2.5×10^6	5.4×10^6	5.2×10^5	1.1×10^4	1.3×10^4	1.3×10^4	3.5×10^{-14}	1×10^5	9.6×10^7	6.3×10^5	7.6×10^5
Minimizing Component Value	27.4	9.2	1	26.7	34.6	1000	37.1	55	9.5	5.3	12.8	26.4

Component	C_1	r_x	r_n	C_u	C_2	R_g	R_e	C_n	C_e	q_m	R_c	R_L
Actual Parameter Value	20	10	40	25	20	75	30	15	30	10	10	20
Minimum Value for J_i	2.6×10^3	2.4×10^3	7.6×10^3	6.1×10^4	3.1×10^3	3.1×10^3	1.3×10^3	7.3×10^2	1.9×10^{-11}	6.7×10^2	2.9×10^3	2.5×10^3
Minimizing Component Value	0.1	1000	1000	10	0.1	0.1	2.7	0.1	30	1000	0.1	0.1

Component	C_1	r_x	r_π	C_u	C_2	R_s	R_e	C_π	C_e	g_m	R_c	R_L
Actual Parameter Value	20	10	40	25	20	75	80	15	10	10	10	20
Minimum Value for J_i	6.8×10^{-5}	5.5×10^{-4}	1.5×10^{-4}	1.5×10^{-4}	4.3×10^{-5}	6.6×10^{-4}	2.2×10^{-4}	8.9×10^{-4}	4.2×10^{-4}	6.4×10^{-4}	1.4×10^{-4}	8.8×10^{-5}
Minimizing Component Value	61.4	8.9	0.4	29.2	1000	1000	80	78.3	9.2	4.6	19.6	47.8

Component	C_1	r_x	r_π	C_u	C_2	R_s	R_e	C_π	C_e	g_m	R_c	R_L
Actual Parameter Value	20	10	40	25	20	75	30	55	10	10	10	20
Minimum Value for J_i	8.5×10^{-5}	2.5×10^{-5}	5.4×10^{-6}	5.2×10^{-5}	1.1×10^{-4}	1.3×10^{-4}	1.3×10^{-4}	3.5×10^{-4}	1×10^{-5}	9.6×10^{-7}	6.3×10^{-5}	7.6×10^{-5}
Minimizing Component Value	27.4	9.2	1	26.7	34.6	1000	37.1	55	9.5	5.3	12.8	26.4

Component	C_1	r_x	r_π	C_u	C_2	R_s	R_e	C_π	C_e	g_m	R_c	R_L
Actual Parameter Value	20	10	40	25	20	75	30	15	30	10	10	20
Minimum Value for J_i	2.6×10^{-3}	2.4×10^{-3}	7.6×10^{-3}	6.1×10^{-2}	3.1×10^{-3}	3.1×10^{-3}	1.3×10^{-3}	7.3×10^{-2}	1.9×10^{-11}	6.7×10^{-2}	2.9×10^{-3}	2.5×10^{-3}
Minimizing Component Value	0.1	1000	1000	10	0.1	0.1	2.7	0.1	30	1000	0.1	0.1

Table 2. Fault Analysis for Transistor Amplifier (Cont:)

Component	C_1	r_x	r_π	C	C_2	R_s	R_e	C_π	C_e	g_m	R_C	R_L
Actual Parameter Value	20	10	40	25	20	75	30	15	10	40	10	20
Minimum Value for J_i	8.9×10^5	9.4×10^6	2×10^4	6.2×10^5	1.1×10^4	1.1×10^4	1.3×10^4	5.7×10^4	2×10^5	5.7×10^5	7×10^5	8×10^5
Minimizing Component Value	15.4	10.9	1000	23.6	13.5	27.6	25.4	0.1	10.5	40	8.1	15.8

Component	C_1	r_x	r_π	C_u	C_2	R_s	R_e	C_π	C_e	g_m	R_C	R_L
Actual Parameter Value	20	10	40	25	20	75	30	15	10	10	50	20
Minimum Value for J_i	6.9×10^5	8.3×10^4	6.7×10^5	5.3×10^5	8.4×10^4	2.3×10^3	3.4×10^4	2.3×10^3	5.1×10^3	1.1×10^4	4.3×10^3	6.2×10^5
Minimizing Component Value	1000	7.8	0.2	33.9	1000	1000	1000	1000	8.4	2.4	50	1000

Component	C_1	r_x	r_π	C_u	C_2	R_s	R_e	C_π	C_e	g_m	R_C	R_L
Actual Parameter Value	20	10	40	25	20	75	30	15	10	10	10	60
Minimum Value for J_i	3.3×10^6	4.2×10^6	4.1×10^5	3.7×10^5	1.3×10^4	9.1×10^4	1.1×10^4	3.9×10^4	2.7×10^4	5.6×10^4	9.1×10^6	4.7×10^6
Minimizing Component Value	84	8.5	0.3	30.3	1000	1000	93.9	1000	8.9	3.6	23.7	60

Table 2. Fault Analysis for Transistor Amplifier (Cont:).

Component	C_1	r_x	r_y	C_u	C_2	R_s	R_e	C_y	C_e	q_m	R_C	R_L
Nominal Parameter Value	20	10	40	25	20	75	30	15	10	10	10	20
Actual Parameter Values	22	11	36	62	22	82	33	14	9	11	9	22
Minimum Value for J_i	7.5×10^{-3}	4.8×10^{-3}	4.7×10^{-3}	2×10^{-5}	1.2×10^{-2}	1.7×10^{-2}	7.9×10^{-3}	3×10^{-3}	2.9×10^{-3}	4.1×10^{-3}	4.2×10^{-3}	6×10^{-3}
Minimizing Component Value	1000	5.6	0.1	69.9	1000	1000	1000	1000	6.4	0.6	1000	1000
Ambiguity Set (X) Actual Fault (0)	■											

Component	C_1	r_x	r_y	C_u	C_2	R_s	R_e	C_y	C_e	q_m	R_C	R_L
Nominal Parameter Values	20	10	40	25	20	75	30	15	10	10	10	20
Actual Parameter Values	22	11	36	27	22	2	33	14	9	11	9	22
Minimum Value for J_i	2.6×10^{-5}	1.3×10^{-3}	7.6×10^{-3}	1.6×10^{-4}	2.6×10^{-6}	2.5×10^{-6}	1.2×10^{-6}	6.7×10^{-3}	6.5×10^{-4}	5.3×10^{-4}	1.1×10^{-4}	5.7×10^{-5}
Minimizing Component Value	4.2	17	1000	17.4	2.52	2.9	10.1	0.1	13.7	1000	2.8	4.9
Ambiguity Set (X) Actual Fault (0)	X ■											

Table 3. Fault Analysis of Transistor Amplifier with 10 percent tolerances

Component	C_1	r_x	r_i	C_u	C_2	R_s	R_e	C_i	C_e	g_m	R_C	R_L
Nominal Parameter Values	20	10	40	25	20	75	30	15	10	10	10	20
Actual Parameter Value	22	11	36	27	22	82	33	37	9	11	9	22
Minimum Value for J_i	7.9×10^{-5}	2.6×10^{-4}	1.4×10^{-4}	1.1×10^{-6}	2.4×10^{-4}	1×10^{-3}	2.2×10^{-3}	2.7×10^{-4}	1.3×10^{-4}	3.6×10^{-4}	3.3×10^{-4}	5.3×10^{-5}
Minimizing Component Value	86.5	8.3	0.3	30.7	1000	1000	94.7	140	8.8	3.1	24.6	62
Ambiguity Set (X)			X	X								
Actual Fault (0)								0				

Table 3. Fault Analysis of Transistor Amplifier with 10 percent tolerances (Cont:).

AN APPROACH TO BUILT-IN TESTING*

R. Saeks

*This research supported in part by Office of Naval Research Contracts 75-C-0924 and 76-C-1136.

I. Introduction

An approach to built-in testing (BIT) for electronic circuits and systems is outlined. The approach assumes a two-level hierarchical architecture in which a central microprocessor controls and coordinates the testing of a number of sub-systems each of which has built-in test equipment (BITE) such as sensors and a nanoprocessor for preprocessing the test data prior to transmission to the central microprocessor. The approach allows for on-line fault detection and prediction up to the level of an SRA (shop replaceable assembly) and off-line fault diagnosis within the various SRA's.

Section II is devoted to a description of the BIT system architecture. This two level structure has been formulated to be applicable either at the printed circuit board level in which the SRA's represent individual devices (IC chips, elementary components, etc.) or at the level of an entire electronics system in which the SRA's represent printed circuit boards.

The third section of the paper is devoted to a study of the fault diagnosis problem. In either the case of a linear or nonlinear circuit it is shown that this problem can be reduced to the solution of a set of simultaneous nonlinear algebraic equations. In the proposed BIT architecture a linearization of these equations is used on-line for fault detection and prediction whereas the full set of nonlinear equations are used off-

line for fault diagnosis within the SRA.

Two algorithms for fault prediction are described in section IV. Both are essentially curve fitting algorithms implemented on the central test microprocessor in a time multiplexed mode. Here the μp periodically receives test data from the various SRA's, and extrapolates this data to determine whether or not the SRA is likely to fail in the near future. The final section of the paper is devoted to a discussion of the concept of self-testing; in particular, the possibility of self testing in a predictive mode.

At the time of this writing the approach to built-in testing described has yet to be fully implemented. It is, however, predicated on several years of research in the area and each of its constituent sub-systems has been extensively simulated^{17,18,19}. At the present time the hardware implementation of the various algorithms is under investigation^{15,16} and we hope to have an entire BIT system in operation in the near future.

II. A BIT Architecture

Our basic BIT architecture is a two-level hierarchical structure illustrated in Figure 1. Intuitively, the overall system may represent a printed circuit board

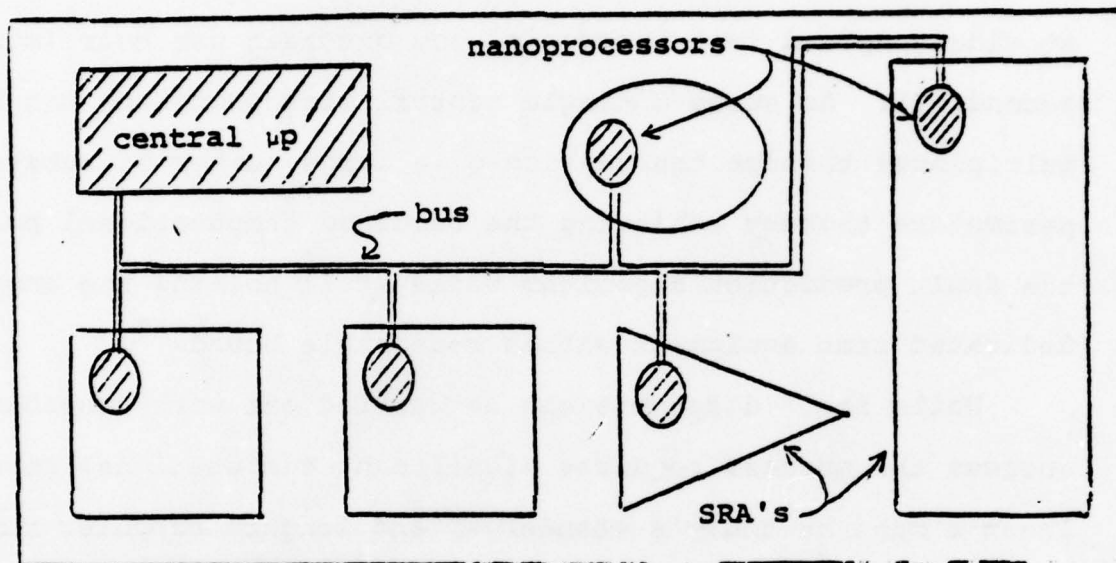


Figure 1: Two-level BIT architecture.

while the subsystems represent various shop replacable assemblies (SRA's) such as integrated circuits, power supplies, SCR's vacuum tubes, etc. Alternatively, the overall system may represent an entire electronics system with the SRA's being its constituent PC boards. In either case the SRA's may be throw-away units or units intended for off-line repair with built-in test equipment (BITE) designed to detect and/or predict faults in the SRA. For those units intended for off-line repair the BITE may also be used as an interface with an external test stand but will not be capable of isolating the failure within the SRA.

This structure is motivated by several years of basic research into the relative computational complexity of the three fundamental problems of fault analysis; fault detection, fault diagnosis, and fault prediction⁹. The latter problem requires considerable computational power^{18,19} but need only be carried out at widely spaced test intervals, say one test per hour (minute, second, ?). As such, a single central microprocessor can be time multiplexed through the testing of a large number of subsystem parameters thereby achieving the required computational power for the fault prediction algorithm while still holding the amount of dedicated test equipment within reasonable bounds¹⁰.

While fault diagnosis can be carried out with considerable success the process requires significant computational power (at least a mini by today's standards) and lengthy computer runs^{7,11,17}. As such, fault diagnosis within an SRA is done off-line on an external test stand containing the required mini (or maxi) computer. Each SRA, however, will include sufficient BITE, say a nanoprocessor, to collect and condition test data on the SRA to be periodically communicated to the central microprocessor for purposes of fault prediction and detection.

Fortunately, both of these endeavors may be achieved using a model of the SRA linearized about its nominal values and hence can be implemented with relatively little computational power built into the SRA¹². For fault prediction in particular, one is interested in tracking various internal parameters of the SRA as they drift from nominal to their tolerance limit. Since the tolerance interval is

typically only a few percent this can be achieved with a linearized model. For catastrophic errors a linearized model may be used to detect failures even though it is not sufficiently accurate for fault diagnosis. As such, the BITE within an SRA may be kept within reasonable bounds while still delivering sufficient data to the central microprocessor for its fault prediction and fault detection tasks. If needed, fault diagnosis within an SRA can, however, be done off-line with the BITE simply serving as an interface between the SRA and an external test stand.

A final aspect of the BIT architecture is the communication link between the SRA's and the central microprocessor. Here, one desires to keep the wiring between the SRA's and the central microprocessor at a minimum and simultaneously have all data transmitted to the central microprocessor in a uniform format to permit interchangeability of component parts within the system. Although the details of this communications link have yet to be formalized the existence of an active computing capability in each SRA gives one considerable flexibility. As such, we believe that it will be possible to work with a single test bus¹⁶. Here, the central microprocessor requests data from the individual SRA's by transmitting a signal on the bus. This signal is received by the built-in nanoprocessor in the SRA which, in turn, transmits appropriately conditioned test data back to the central microprocessor on the same bus.

The above described BIT architecture would seem to achieve most of the requirements for a built-in testing system.

- i). Continuous on-line fault prediction and detection to an SRA is achieved.
- ii). The system includes an interface for off-line fault diagnosis within an SRA.
- iii). Dedicated test equipment represents a small percentage of the total system.
- iv). Busing is minimized and test data is transmitted to the central microprocessor in a uniform format thereby facilitating component interchangeability.

III. Fault Diagnosis

For the purposes of doing fault diagnosis we work with a component connection model for the circuit or system under test which takes for form

$$\begin{aligned} 1. \quad & b_i = Z_i(s, r) a_i \quad i=1, 2, \dots, n \\ \text{and} \quad & a = L_{11}b + L_{12}u \\ 2. \quad & y = L_{21}b + L_{22}u \end{aligned}$$

in the frequency domain,^{6,11,12}. Here $Z_i(s, r)$ is the transfer function of the i th circuit or system component where $R = \text{col}(r_i)$ is the vector of unknown component parameters and s is the complex frequency variable. Typically, the unknown component parameters take the form of amplifier gains and cut-off frequencies, pole and zero positions, resistances, inductances, etc. In particular, it is assumed that enough parameters are employed to completely characterize the performance of the device. The L_{ij} are known connection matrices, $a = \text{col}(a_i)$ and $b = \text{col}(b_i)$ are composite vectors of component inputs and outputs respectively, and u and y are the test input and output signals respectively. In the nonlinear case the component equations are replaced by the state models

$$\begin{aligned} 3. \quad & \dot{X}_i = f_i(X_i, a_i, r) \\ & \quad \quad \quad ; i=1, 2, \dots, n \\ & b_i = g_i(X_i, a_i, r) \end{aligned}$$

with the connection equations remaining as in 2. Although these component connection models for a circuit or system are non-classical they are widely used in large-scale system simulation and computer-aided circuit design and are readily amenable to the

"computer speed-up techniques" developed for these applications¹². As such, they are ideally suited for the fault diagnosis problem.

Combining 1. and 2. yields the fault diagnosis equation⁷.

$$4. \quad S^m = L_{22} + L_{21}(1 - Z(s, r)L_{11})^{-1}Z(s, r)L_{12}$$

where $Z(s, r) = \text{diag}(Z_i(s, r))$ and S^m is the measured transfer function relating the input test signal u to the output test signal y . The solution of the fault diagnosis problem therefore amounts to the solution of 4. for the parameters vector, r , given S^m and the connection matrices. Although it is possible to give an analytic description of all possible solutions to this equation^{12,13} given any fixed value for the complex frequency variable, s , in a "real world" situation the number of unknowns greatly exceeds the number of equations and, as such, the analytic representation of the solution manifold proves to be of little value. This difficulty is alleviated via a multi-frequency diagnosis algorithm wherein one writes the set of simultaneous equations

$$\begin{aligned} 5. \quad S(s_1, r) &= L_{22} + L_{21}(1 - Z(s_1, r)L_{11})^{-1}Z(s_1, r)L_{12} \\ S(s_2, r) &= L_{22} + L_{21}(1 - Z(s_2, r)L_{11})^{-1}Z(s_2, r)L_{12} \\ &\vdots \\ S(s_k, r) &= L_{22} + L_{21}(1 - Z(s_k, r)L_{11})^{-1}Z(s_k, r)L_{12} \end{aligned}$$

where k different complex frequencies are used in equation 4. simultaneously. The interesting and somewhat surprising result is that the additional equations in 5. may be independent thus increasing the number of fault diagnosis equations without increasing the number of its unknowns⁷. While the set of simultaneous equations 5, often has a unique solution, no analytic solution technique is known and we must resort to time consuming numerical solution procedures carried out off-line.

Although the multi-frequency fault diagnosis equations of 5. do not admit an analytic solution their numerical solution can be significantly speeded up by careful analysis of the equations. In particular, a little algebra^{6,12} will reveal that

$$6. \quad \frac{ds^m}{dr_j} = L_{21}(1 - Z(s_i, r)L_{11})^{-1} \left[\frac{dz(s_i, r)}{dr_j} \right] [1 + L_{11}(1 - Z(s_i, r)L_{11})^{-1}] L_{12}$$

showing that one can compute the partial derivatives required for the numerical solution to 5. analytically. Moreover, if one observes that the inverse matrix required to compute the partial derivatives in equation 6, is precisely the same inverse matrix required to evaluate the multi-frequency fault diagnosis equations 5. it is seen that the partial derivative information is obtained at virtually no computational cost over and above that required for the evaluation of the equations. In a similar vein one can reduce the computation required to compute the inverses at different complex frequencies by integrating the differential equation

$$7. \quad \frac{d(1-Z(s,r)L_{11})^{-1}}{ds} = (1-Z(s,r)L_{11})^{-1} \left[\frac{dZ(s,r)}{ds} \right] L_{11} (1-Z(s,r)L_{11})^{-1}$$

using the inverse computed at one particular frequency as a starting point^{2,14}. Although of extremely high dimension this equation is easily integrated without the requirement for matrix inversions. With the aid of these observations it is possible to carry out an entire iteration of a Newton-Raphson algorithm for the solution of the multi-frequency fault diagnosis equations with the aid of only a single matrix inversion.

Although one does not have a "neat" set of equations such as those described above for the solution of the fault diagnosis problem in a nonlinear circuit or system, surprisingly similar computational techniques can be invoked in the nonlinear case. The key to these techniques is the replacement of the multi-frequency information of the linear case by a family of integral performance measures on the test signals, u and y . These play exactly the same role in nonlinear fault diagnosis as played by the frequency information in the linear case, allowing one to formulate multiple independent fault diagnosis equations from the same test signals.

In the nonlinear case, the sparse tableau algorithm^{3,12} is used to evaluate the fault diagnosis equations at each iteration of a Newton-Raphson algorithm. As in the linear case this algorithm allows one to compute the derivative required for the Newton-Raphson algorithm with essentially no

additional computational cost over and above that required for the evaluation of the equations^{3,4,12}. It is possible to obtain significant computational gains in the solution of the fault diagnosis equations in the nonlinear case as well as the linear case, by optimally exploiting the computational efficiencies inherent in the sparse tableau formulation for an electronic circuit or system.

Even using computational efficiencies which are possible for solving the fault diagnosis equations, this method is still long and tedious and not well suited to on-line implementation in a BIT system. It is thus recommended that linearization of the fault diagnosis equations be used instead. Although far less accurate than the solution of the full set of fault diagnosis equations¹², we believe that in the context of the previously described BIT architecture linearization of the fault diagnosis equations will prove to be viable. From the point of view of fault prediction one is interested only in tracking the unknown parameter vector r , from its nominal value to its tolerance limit, (a few percent deviation from nominal). This is a region in which the solution of the linearized fault diagnosis equations should be quite accurate. On the other hand, if a catastrophic fault occurs, solution of the linearized equations will detect the fault though it may fail to accurately diagnosis it. In this case, however, the linearized test data and its associated BITE may be employed as an interface between

the SRA and an external test stand. As such, the use of linearized fault diagnosis equations will suffice in the context of our BIT architecture.

From the point of view of on-line analysis in a BIT system the solution of the linearized fault diagnosis equations is computationally reasonable. Since the linearization is done about the nominal value, it may be precomputed (via equation 6. in the linear case and the corresponding equation in the nonlinear case) and its inverse may be precomputed. Thus, the implementation of an algorithm for the solution of the linearized fault diagnosis equations requires only a single matrix multiplication, the matrix having been precomputed off-line and stored in a ROM.

IV. Fault Prediction

In the context of the previously described BIT architecture the primary role of the central microprocessor is to periodically collect data from the individual SRA's characterizing their internal parameter vectors, r . This data is then used to detect and predict failures of the SRA. When a failure is detected, the central microprocessor signals this fact and the SRA is replaced and/or taken to an external test stand for repair. If no failure is detected, the role of the central microprocessor is to compare the present data with previously measured values in an endeavor to predict whether or not failure is imminent. In this instance predicted failure of the SRA would be signaled in a effort to replace the device before its actual on-line failure.²⁰

For any particular device one can collect statistical data on which to base a fault prediction algorithm. However, in a practical BIT setting where the same fault prediction algorithm is multiplexed through the testing of many different SRA's, it is necessary to use an algorithm which is independent of the specific properties of the parameter under test. As such, for our BIT system, we expect to employ a curve fitting algorithm²⁰. Although less accurate than a statistically based algorithm, we have shown by simulation^{18,19}, that such an algorithm can be employed as a satisfactory fault predictor. Such algorithms are computationally simple thus permitting a single central microprocessor to be multiplexed through the

testing of a large number of SRA's¹⁵. Moreover, if one assumes that the data delivered by the SRA to the central microprocessor has been uniformly normalized, the fault prediction algorithm will be completely independent of the parameter under test. As such, one is in a position to completely standardize the central microprocessor in a BIT system so that changes in an SRA do not demand corresponding changes in the fault prediction algorithm.

Over the past several years we have investigated several approaches to the fault prediction problem^{5,15,16,18,19,20}. The first is extremely naive but has yielded surprisingly effective results in simulation^{18,19,20}. Basically, one collects data at periodic intervals, fits the data with a second order polynomial, and solves the quadratic equation to estimate the time at which the parameter will go out of tolerance. The success of this algorithm is due to the fact that one is not really interested in the accuracy of the failure time estimate but only the accuracy of the binary decision (based on this estimate) whether or not to replace the SRA. Moreover, this binary decision is only made when failure is expected in the near future, a region of time in which a polynomial extrapolation is reasonably accurate. I.e., if failure is estimated to take place in 3 years even if the estimate is off by 90%, the decision not to replace the SRA at this time will still be correct.

A fault prediction algorithm based on the above described second order polynomial extrapolation has been exten-

sively studied by Tung and this author on some 10,000 complete simulated operations of the algorithm^{18,19}. Most of these simulations were carried out on artificial data generated by a library of special functions to which a noise term was added. These special functions included some highly complex non-monotonic curves. Additionally, curves based on the empirical drift formula for thin film resistors were studied ($R(t) = At^a$ where a lies between .3 and .5)^{18,19}. In both cases, random noise with amplitudes of up to 25% of the tolerance interval was added to the data. The result of these simulations, which we believe to represent an environment which is more extreme than the "real world", was that 99.5% of all SRA's were replaced before on-line failure at a cost of about 10% of their lifetime.

At the present time a somewhat more sophisticated fault prediction algorithm is under development⁵. This is still essentially a curve fitting algorithm though one in which a failure model (founded in modern reliability theory¹) is employed. The basic idea for this algorithm is as follows. The drifting SRA parameter, r , is assumed to satisfy a difference equation

$$8. \quad r(k+1) = r(k) + f(k)$$

where the "component time" k represents the number of shocks the SRA has received (eg. switching processes, electrons boiling off a cathode, etc) The relation between component time, i.e. the number of shocks received, and "real time" is assumed to be a Poisson distributed random variable in which the

probability of the SRA receiving n shocks in a time interval of length t is

$$9. \quad P_n(t) = (ct)^n e^{-ct} / \underline{n}$$

It is assumed that the value of the parameter r is known for a fixed set of points in "real time"; $r(t_1), r(t_2), \dots, r(t_m)$. Using this data we desire to estimate the unknown failure dynamics, $f(k)$, for the SRA parameter. This is then used in equation 8. to compute the number of shocks required to cause failure; i.e. the smallest value of k for which $r(k)$ is out of tolerance. Finally, this estimate is used to compute the optimal "real time" at which to replace the SRA to minimize the cost functional

$$10. \quad J = c_f P_f + c_w W$$

Here, P_f is the probability of on-line failure, W is the average percentage of SRA lifetime which is wasted by replacing the SRA before its actual failure and c_f and c_w are weighting factors.

Note that the implementation of the above described Poission shock based fault prediction algorithm requires that we deal simultaneously with two unknown phenomena: the failure dynamics, $f(k)$, and the random relationship between "real time" and "component time" given by the Poission distribution. Although the required analysis is complex a surprisingly tractable (and optimal in an appropriate sense) fault prediction algorithm can be formulated.

The properties of the Poission distribution are used to estimate the number of shocks which the SRA has received in the time intervals $[t_i, t_{i-1}]$; $i=1,2,\dots,m$ in combination with a generalized inverse algorithm to estimate $f(k)$. $f(k)$ is approximated by a j th order polynomial and one must compute the generalized inverse of an m by j matrix. Fortunately, the algorithm is ideally suited to a sequential least squares technique⁸ and no matrix inversions need be carried out on-line. Once $f(k)$ has been estimated to a satisfactory level of accuracy (by increasing the order of the approximating polynomial until the estimation error is reduced to a prescribed level) it is used with equation 8. to compute the number of shocks, required for the parameter to go out of tolerance. Finally, this value is used in conjunction with the Poission distribution to determine the optimal "real time" at which to replace the SRA. Although apparently complex, this latter optimization can be reduced by analytic techniques to the solution of a single nonlinear equation in one variable⁵. As such, the entire fault prediction algorithm may be easily implemented, on-line, in a BIT system. Unlike the second order curve fitting algorithm the Poission shock algorithm for fault prediction is still under development and its simulation on "real world" data, is just beginning.

From the point of view of our BIT architecture where the central microprocessor is dedicated exclusively to the fault prediction job (plus bookkeeping and control of the test communications link), we anticipate little difficulty in implementing either of our fault prediction algorithms. The key to the viability of the concept, however, is to make the algorithm fast enough so that a single central micro-

processor can be time-multiplexed to test a large number of parameters. In an effort to verify the feasibility of such an approach, we are presently in the process of implementing the second order curve fitting algorithm on an F8 micro-processor¹⁵. Although the implementation has yet to be completed, most of the sub-programs have been written and tested and it appears that the program will require about 500 bytes of memory and execute in about 30 milliseconds. As such, the central microprocessor should be able to cycle through the testing of about 2000 parameters at one minute intervals.

V. Self Testing

An interesting side effect of running a BIT system in a predictive mode is that it opens the possibility of reliable self testing. The key observation is that to do fault prediction in a digital device one must test analog parameters such as rise time, power supply voltage, clock rate, pulse widths, etc. since digital parameters are either right or wrong and have no gray region from which to extrapolate trends. One may, therefore, use a microprocessor to predict its own failure by extrapolating the values of its analog parameters. As long as the prediction is made before these parameters go out of tolerance and the digital performance of the microprocessor is still exact. The point is that in a predictive mode the microprocessor is still working at the time it predicts its own failure and hence may be used reliably in a self testing mode. Of course, once the analog parameters of the microprocessor have exceeded their tolerance limits, it may no longer be trusted as a digital signal processor and hence the device cannot be used to diagnose its own faults after failure.

Although the above described self testing concept is purely conceptual and has yet to be implemented or even simulated, it is indicative of the potential of fault prediction in a BIT system. Indeed, if one can reliably predict failure before it actually takes place, such concepts as self repair move into the realm of feasibility, since at the time a replacement decision is made the device under test is still working.

VI. Conclusions

Our purpose in the preceding has been to outline an approach for designing a built-in test system applicable to electronic circuits and systems. Although not yet implemented in hardware, each of the constituent parts of the BIT system have been extensively simulated and we believe that a hardware implementation is feasible both computationally and economically. At the present time, we are implementing the polynomial curve fitting algorithm for fault prediction in hardware and are in the preliminary stages of implementing the entire system in a high voltage power supply.

VI. References

1. Barlow, R.E., and F. Prochan, Statistical Theory of Reliability and Life Testing: Probability Methods, New York, Holt, Rinehard and Winston, 1975.
2. Chao, K.-S., and R. Saeks, "Continuation Methods in Circuit Analysis", IEEE Proc., v. 65, pp. 1187-1194, (1977).
3. Hachtel, G.D., Brayton, R.K., and F.G. Gustavson, "The Sparse Tableau Approach to Network Analysis and Design", IEEE Trans. on Circuit Theory, Vol. CT-18, pp. 101-113, (1971).
4. Hachtel, G.D., and R.A. Rohrer, "Techniques for the Optimal Design Synthesis of Switching Circuits", IEEE Proc., Vol. 55, pp. 1864-1877, (1967).
5. Lu, K.-S. Ph.D. Thesis, Texas Tech University., Lubbock, Texas., (1977).
6. Ranson, M.N., and R. Saeks, "The Connection Function - Theory and Application", Int. Jour. of Circuit Theory and its Application, Vol. 3, pp. 5-21, (1975).
7. Ransom, M.N., and R. Saeks, "A Functional Approach to Fault Analysis", in Rational Fault Analysis, New York, Marcel Dekker Inc., pp. 124-134, (1977).
8. Rao, C.R., and S.K. Mitra, Generalized Inverse of Matrices and its Applications, New York, J. Wiley and Sons, 1971.
9. Saeks, R., etal., "Fault Analysis in Electronic Circuits and Systems", Tech. Rpt., Texas Tech Univ., Lubbock, Texas., Nov. 1975.
10. Saeks, R., "An Experiment in Fault Prediction II", Proc. of the 1976 IEEE AUTOTESTCON, Arlington, Tx., Nov. 1976, p. 53, (abstract only).
11. Saeks, R., Singh, S.P., and R.-w. Liu, "Fault Isolation via Components Simulation", IEEE Trans. on Circuit Theory, Vol. CT-19, pp. 634-640, (1972).
12. Saeks, R., and R.A. DeCarlo, "Interconnected Dynamical Systems", Marcel Dekker Inc., (to appear).
13. Saeks, R., Wise, G., and K.-S. Chao, "Analysis and Design of Interconnected Dynamical Systems", in Large-Scale Dynamical Systems, No. Hollywood, Point Lobos Press, 1976, pp. 59-79.

14. Saeks, R., and K.-S. Chao, "Continuations Approach to Large Change Sensitivity Analysis", IEEE (London) Jour. on Electronic Circuits and Systems, Vol. 1, pp. 11-16, (1976).
15. Sangani, S., Unpublished Notes, University of Detroit, Detroit, Mi., 1976.
16. Sen, N., M.S. Thesis, Texas Tech University, Lubbock, Texas, 1975.
17. Tsui, Y.-L., M.S. Thesis, Texas Tech University, Lubbock, Texas, 1975.
18. Tung, L., M.S. Thesis, Texas Tech University, Lubbock, Texas, 1975.
19. Tung, L., and R. Saeks, "An Experiment in Fault Prediction", Proc. of the 4th Symp. on Reliability in Electronics, Budapest, Oct. 1977, pp. 249-257.
20. Tung, L., Liberty, S.R., and R. Saeks, "Fault Prediction - Towards a Mathematical Theory", in Rational Fault Analysis, New York, Marcel Dekker, pp. 135-142, (1977).

FAILURE PREDICTION FOR AN ON-LINE MAINTENANCE SYSTEM IN A
POISSON SHOCK ENVIRONMENT*

K.S. Lu** and R. Saeks

*This research supported in part by Office of Naval Research Contracts
75-C-0924 and 76-C-1136.

**Presently with the Texas Instruments Corporation, Dallas, Texas.

I. Introduction

Fault analysis processes, have been and will continue to be very significant factors in the safety and reliability of electrical systems. This is especially true due to the following facts: a rapid advancement in the complexity and size of modern systems, increased availability and capabilities of computers, and rapidly changing technologies in integrated circuit fabrication. Due to this, fault analysis has become much more than an academic research topic. Fault analysis is applicable in an industrial environment to minimize cost, extend the lifetime of the overall system, control maintenance schedules, and effectively plan manpower needs.

Although considerable effort has been expended during the past decade to develop techniques for fault detection and diagnosis in both analog and digital electronic circuits¹ little attention has been given to the possibility of formulating algorithms for fault prediction. To accurately predict a fault, a device must be tested at periodic maintenance intervals. If the device fails or does not operate correctly, it is replaced immediately. The device may be assumed good if its characteristics are in tolerance. However, if the characteristics are slightly off nominal, but the device still operates correctly, one can attempt to predict if the device will fail before the next scheduled maintenance interval. If device failure is predicted, it can be replaced before failure occurs as part of planned preventative maintenance.

With the advent of the low-cost microprocessor, on-line fault

prediction is possible and practical.² A curve fitting algorithm for on-line fault prediction was first introduced by Saeks, Liberty and Tung^{3,4,5} in 1975. It was assumed that prior life-time statistics for the system under test were known. Also, performance data of the system at each maintenance interval were collected. The application of these data to a second order polynomial equation resulted in an estimation of the time at which the component under test would exceed tolerance limits. Based on a criterion of simultaneously minimizing on-line failures and maximizing component lifetime, a decision as to whether or not the component should be replaced is made at each maintenance interval.

The disadvantages of this curve fitting algorithm are: the application is limited to failures due to permanent overstress, the second order polynomial is too simple to describe the performance of the component, and the prior lifetime statistics for the component are often not available.

Another area where an extensive research effort is being applied is shock models and wear processes. Esary, Marshall and Proschan^{6,7,8} introduced a shock model for the life distribution of a component subjected to a sequence of shocks randomly occurring in time according to a homogeneous Poisson process. They also considered the related shock models in which each shock caused a random amount of damage and failure occurred when the accumulated damage exceeded a specified threshold. This failure model is well known in modern reliability theory.

Employing the Poisson-Shock model, another curve fitting fault prediction algorithm which will overcome the disadvantages of the

Saeks-Liberty-Tung algorithm will be discussed in the present paper.⁹ In the following section, a model for the failure dynamics of a system component parameter is formulated. Here, it is assumed that the failure is due to the component being subjected to a sequence of Poisson distributed shocks^{10,11} with the measurable parameter being controlled by an unknown difference equation whose underlying discrete "component time" process is defined by the number of shock to which the component has been subjected. Since both the failure dynamics (i.e. the difference equation) and the relationship between "component time" and real time are unknown, our failure model is doubly stochastic. The third section of the paper is devoted to the formulation of an algorithm for estimating the component failure dynamics and its "lifetime", defined to be the number of shocks required to cause component failure. This is followed by the formulation of an "optimal" replacement theory wherein the optimal real time at which to replace a component is computed in terms of its estimated "lifetime". Finally, the results of a simulation of the algorithm in both an ideal and noisy environment are presented and compared with the simulated performance for several fixed replacement schedules.

II. Failure Dynamics

The performance of an analog device subject to a series of discrete shocks (switching process, improper operation, etc...) may drift due to the shock damage. Let $C(N)$ represent values of a particular component parameter, where the "component time", N , denotes the number of shocks the component has received. It is assumed that the drifting parameters can be described by a first

order[†] difference equation of the form:

$$1. \quad C(N+1) = C(N) - a_0 - a_1 N - a_2 N^2 - \dots - a_h N^h \quad C(0) = 1$$

Here, the coefficients and order of the "forcing polynomial" are assumed to be unknown and must be estimated as part of the fault prediction process. A little algebra together with the standard recursive formula for solving a difference equations will reveal that

$$2. \quad C(N) = 1 - \sum_{j=0}^{N-1} \sum_{i=0}^h a_i j^i$$

Now, if the tolerance limit for the component parameter is taken to be $C = 0$, we may define the lifetime of the component to be the smallest integer, N , for which $C(N) \leq 0$. This integer which we denote by L then represents the number of shocks necessary to cause the component to fail.

Consider a simple example where the "forcing polynomial" is taken to be of the first order with positive coefficients. Then 1. reduces to

$$C(N+1) = C(N) - a_0 - a_1 N ; \quad C(0) = 1$$

From equation 2. $C(N)$ can thus be expressed as

$$C(N) = 1 - a_0 N - \frac{N(N-1)}{2} a_1$$

[†]The concepts described herein carry over without modification to the case where the failure model is characterized by higher order difference equations. The first order model, however, suffices to illustrate the theory and is hence used throughout the present paper.

Then the lifetime of this component is the smallest integer satisfying the equation

$$1 - a_0 L - \frac{L(L-1)}{2} a_1 \geq 0$$

That is, L is the smallest integer such that

$$L \geq \frac{(2a_0 - a_1)^2 + 8a_1 - (2a_0 - a_1)}{2a_1}$$

Since the failure model of equation 1. is dependent on "component time", i.e. the number of shocks the component has received, rather than real time, it remains to define the relationship between "component time" and real time. Following common practice in reliability theory⁶, we assume that this relationship is determined by a Poisson process. Indeed, this is the unique point process which has the scaling properties required for such an application.¹¹ Here, the probability of N shocks occurring in the time interval t is:

$$3. \quad P_N(t) = e^{-kt} \frac{(kt)^N}{N!} \quad N = 0, 1, 2, \dots$$

Where k is a given constant representing the average number of shocks per unit time. Therefore, (kt) is the average number of shocks in the time interval t .

If a component with lifetime L is subjected to Poisson shock with constant k the probability that it will fail (i.e. receive at least L shocks) by time t , is then given by the formula⁹

$$\begin{aligned}
 F(t) &= \sum_{n=0}^{L-1} P_n(t) \\
 &= \sum_{n=0}^{L-1} e^{-kt} \frac{(kt)^n}{n!}
 \end{aligned}$$

Thus, even though the lifetime of our component is integer valued, in our model the actual failure time is continuously distributed since the time at which the component receives the L th shock is continuously distributed.

III. Estimation of Failure Dynamics and Lifetime

In a periodic maintenance system, the performance of a component is measured at each maintenance interval nT . That is to say, (C_1, C_2, \dots, C_g) is the performance data taken at maintenance times $(T, 2T, \dots, gT)$. The estimation problem can be stated as:

"Given performance data (C_1, C_2, \dots, C_g) , T and k , estimate the unknown constants (a_0, a_1, \dots, a_h) of the failure dynamics." Since it is assumed that the system is subjected to Poisson Shock with constant k , the expected number of shocks in each maintenance interval is kT .[†] As such, if we assume that C_m is the value of the component parameter at $N = mkT$, then upon substituting $C_m = C(mkT)$ into equation 2. we obtain

$$\sum_{j=0}^{mkT-1} a_0 j^0 + \sum_{j=0}^{mkT-1} a_1 j^1 + \dots + \sum_{j=0}^{mkT-1} a_h j^h = 1 - C_m$$

where $m = 1, 2, 3, \dots, g$ or in the matrix form:

[†]Although not theoretically necessary, we assume that kT is an integer.

$$4. \quad JA \triangleq \begin{bmatrix} \sum_{j=0}^{kT-1} j^0 & \sum_{j=0}^{kT-1} j^1 & \dots & \sum_{j=0}^{kT-1} j^h \\ \sum_{j=0}^{2kT-1} j^0 & \sum_{j=0}^{2kT-1} j^1 & \dots & \sum_{j=0}^{2kT-1} j^h \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{j=0}^{gkT-1} j^0 & \sum_{j=0}^{gkT-1} j^1 & \dots & \sum_{j=0}^{gkT-1} j^h \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_h \end{bmatrix} = \begin{bmatrix} 1-C_1 \\ 1-C_2 \\ \vdots \\ 1-C_g \end{bmatrix} \triangleq Z$$

Since the number of data points, g , is typically much greater than the order of the polynomial assumed in the failure model, h , it is not expected that equation 4. admits an exact solution. Rather, we attempt to solve for a coefficient vector, A , which minimizes the error between JA and Z . In particular, if one adopts a least squares error criterion the optimal A is given by

$$5. \quad A^0 = J^{-G}Z$$

where J^{-G} denotes the generalized inverse of J .¹² Indeed, if as is typically the case J has full column rank then $J^{-G} = (J^t J)^{-1} J^t$ where "t" denotes matrix transposition. As such, we take the $A^0 = \text{col}(a_0^0, a_1^0, \dots, a_h^0)$ as our estimate of the coefficients of the difference equation characterizing the failure dynamics of our drifting parameter, C , as per equation 1.

To estimate the failure dynamics of a drifting parameter, the proper choice of the order h is, in general, quite difficult and depends upon physical considerations and engineering experience.

Once h is preselected, however, coefficients to best approximate the failure dynamics can be readily computed via equation 5. The accuracy of the resultant estimate, however, is highly dependent on the choice of the order, h , and on the number of measurements which are taken, g . To find a new set of coefficients for a different combination of h and g , the entire calculation procedure is typically repeated from the very beginning which is impractical in the on-line maintenance system. Fortunately, sequential refinement schemes for obtaining new sets of coefficients without repeating the entire calculation can be developed.^{12,13} As such, it is possible to sequentially update one's estimates of the parameters; a_0, a_1, \dots, a_h ; as additional measurements are taken and/or to increase the order of the model for the failure dynamics without repetitious matrix inversion. Our algorithm for estimation of the failure dynamic underlying the measured data may thus be readily implemented on-line with the computational power presently available in today's microprocessors. The matrix algebraic details of the required sequential refinement schemes are straightforward^{12,13} and readily available in the literature. As such, they will not be repeated here.

In practice, given g measurements C_1, C_2, \dots, C_g taken at maintenance intervals $T, 2T, 3T, \dots, gT$, one sequentially estimates the coefficients of the failure dynamics; a_0, a_1, \dots, a_h ; increasing h until no further error reduction is achieved. The resultant set of coefficients is then used in equation 2. to determine the component lifetime, L . Upon solving the equation,

the resultant estimated lifetime is found to be the smallest integer, L , such that

$$6. \quad \sum_{j=0}^{L-1} \sum_{i=0}^h a_i j^i \geq 1$$

Of course, if the measured data is not decaying towards zero, i.e. the component isn't failing, this inequality will have no solution in which case we take L to be infinite.⁹

IV. Replacement Theory

Although the algorithm outlined in the preceeding section yields an "optimal" estimate of the number of shocks required to cause failure the time at which the L th shock takes place is statistical in nature and hence, it still remains to determine the optimal (in an appropriate sense) time at which to replace the component. One such criterion is formulated in the following. For this purpose, it is assumed that L has been computed to our satisfaction and we desire to choose a time, T_r , at which to replace the component as a function of L . Given L and T_r we denote the resultant probability of on-line failure (i.e. failure before T_r) by P_f . $P_r = 1 - P_f$ then denotes the probability that the component is replaced at time T_r before it fails. Similarly, we let \hat{T}_f denote the expected time to failure for those components which fail on-line, we let \hat{T} denote the expected time to failure for all components and we let T^* denote the expected time to failure for the components if they were operated to failure without replacement (i.e. $T^* = \hat{T}|_{T_r \rightarrow \infty}$). Finally, we let $f_L(t)$ denote the probability density function that the component receives the L th shock at time t , given that the component fails on-line, whereas, $p_i(t)$ represents the density function of the Poisson distribution with parameter (k) and $E_L(t)$ represent the

corresponding distribution function; i.e.

$$7. \quad p_i(t) = \frac{(kt)^i}{i!} e^{-kt} \quad i = 0, 1, 2, \dots$$

$$8. \quad E_L(t) = \sum_{i=0}^{L-1} p_i(t)$$

With the aid of some elementary calculus⁹ P_f, P_r, \hat{T}_f , and \hat{T} , as well as their derivatives with respect to T_r , can be computed analytically. As such, upon defining an appropriate cost measure an explicit formula for determining an "optimal" T_r given L can be derived. We begin with the derivation of the explicit formula for the various quantities involved in our replacement theory.

Since a component will be replaced by our algorithm if and only if it is still operating at time T_r , i.e. if it has not yet received L shocks at time T_r , the probability of replacement is just the probability of receiving less than L shocks by time T_r . We thus have:

$$\begin{aligned} (\text{prop 1}) \quad P_r &= E_L(T_r) \\ P_r &= \sum_{i=0}^{L-1} \frac{(kT_r)^i}{i!} e^{-kT_r} \\ &= \sum_{i=0}^{L-1} p_i(T_r) \\ &= E_L(T_r) \end{aligned}$$

$$(\text{prop 2}) \quad P_f = 1 - E_L(T_r)$$

$$(\text{prop 3}) \quad \int_0^{T_r} p_i(t) dt = \frac{1}{k} \{1 - E_{i+1}(T_r)\}$$

$$\begin{aligned} \int_0^{T_r} p_i(t) dt &= \int_0^{T_r} \frac{(kt)^i}{i!} e^{-kt} dt \\ &= \frac{k^i}{i!} \int_0^{T_r} t^i e^{-kt} dt \end{aligned}$$

Using the identity

$$\int x^m e^{ax} dx = e^{ax} \sum_{r=0}^m (-1)^r \frac{m! x^{m-r}}{(m-r)! a^{r+1}}$$

this becomes

$$\begin{aligned} \int_0^{T_r} p_i(t) dt &= \frac{k^i}{i!} e^{-kt} \sum_{r=0}^i (-1)^r \frac{i! t^{i-r}}{(i-r)! (-k)^{r+1}} \Bigg|_0^{T_r} \\ &= \frac{k^i}{i!} \left\{ e^{-k \cdot 0} \frac{i!}{k^{i+1}} - e^{-kT_r} \sum_{r=0}^i \frac{i! T_r^{i-r}}{(i-r)! k^{r+1}} \right\} \\ &= \frac{1}{k} \left\{ 1 - e^{-kT_r} \sum_{r=0}^i \frac{(kT_r)^{i-r}}{(i-r)!} \right\} \\ &= \frac{1}{k} \left\{ 1 - e^{-kT_r} \sum_{j=0}^i \frac{(kT_r)^j}{j!} \right\} \\ &= \frac{1}{k} \left\{ 1 - \sum_{j=0}^i p_j(T_r) \right\} \\ &= \frac{1}{k} \left\{ 1 - E_{i+1}(T_r) \right\} \end{aligned}$$

$$\text{(prop 4)} \quad f_L(t) = \frac{P_{L-1}(t)}{1/k (1-E_L(T_r))}$$

To derive this conditional density function we partition the interval $(0, T_r)$ into N segments of length $\Delta = T_r/N$ and we compute the probability that the L th shock takes place in the i th time interval, $((i-1)\Delta, i\Delta]$. Since, this can be caused by having $L-1$ shocks before $(i-1)\Delta$ and at least one shock in the interval $((i-1)\Delta, i\Delta]$ or by having $L-2$ shocks before $(i-1)\Delta$ and at least two shocks in the interval $((i-1)\Delta, i\Delta]$, etc. the probability of failure in the i th interval is given by

$$\begin{aligned} & \sum_{j=1}^L P_{L-j}((i-1)\Delta) [1 - E_j(\Delta)] \\ 9. \quad & = \sum_{j=1}^L P_{L-j}((i-1)\Delta) \sum_{q=j}^{\infty} \frac{(\Delta k)^q}{q!} e^{-\Delta k} \\ & = \sum_{r=1}^{\infty} \frac{1}{r!} \left[\sum_{s=1}^r P_{L-s}((i-1)\Delta) \right] (\Delta k)^r e^{-\Delta k} \end{aligned}$$

Taking the probability density function at a point t in the interval $((i-1)\Delta, i\Delta]$ to be limiting value of this quantity divided by Δ as Δ goes to zero¹⁰ it is observed that the terms of 9. containing powers of (Δk) greater than 1 go to zero in the limit. As such, the probability density function for the L th shock to take place at time t is given by

$$10. \quad \lim_{\Delta \rightarrow 0} \frac{P_{L-1}((i-1)\Delta) (\Delta k) e^{-k\Delta}}{\Delta} = k P_{L-1}((i-1)\Delta)$$

AD-A060 941

TEXAS TECH UNIV LUBBOCK INST FOR ELECTRONICS SCIENCE
FAULT ANALYSIS IN ELECTRONIC CIRCUITS AND SYSTEMS, II.(U)
JAN 78 R SAEKS, N SEN, H M CHEN, K S LU

F/G 9/3

N00014-75-C-0924

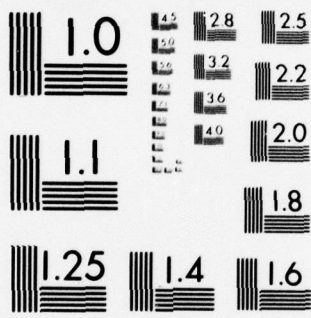
UNCLASSIFIED

NL

2 OF 4

AD
A060941





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Finally, since we are only interested in the conditional probability density function that the L th shock will take place at time t given that the component fails on-line the quantity of equation 10. must be normalized yielding

$$f_L(t) = \frac{kp_{L-1}((i-1)\Delta)}{P_f} = \frac{p_{L-1}(t)}{\frac{1}{k} (1-E_L(T_r))}$$

as was to be shown.

$$(\text{Prop 5}) \quad \tilde{T}_f = \frac{L}{k} \frac{1 - E_{L+1}(T_r)}{1 - E_L(T_r)}$$

Since T_f is the expected lifetime of the components which fail before replacement,

$$\begin{aligned} T_f &= \int_0^{T_r} t f_L(t) dt \\ &= \int_0^{T_r} t \frac{p_{L-1}(t)}{\frac{1}{k} (1-E_L(T_r))} dt \\ &= \frac{\int_0^{T_r} t \frac{(kt)^{L-1}}{(L-1)!} e^{-kt} dt}{\frac{1}{k} (1 - E_L(T_r))} \\ &= \frac{\frac{L}{k} \int_0^{T_r} \frac{(kt)^L}{L!} e^{-kt} dt}{\frac{1}{k} (1-E_L(T_r))} \\ &= \frac{L \int_0^{T_r} p_L(t) dt}{(1 - E_L(T_r))} \end{aligned}$$

From (prop 3), equation 11. thus reduces to the desired equality.

$$(\text{Prop 6}) \quad \hat{T} = \frac{L}{k} \{1 - E_{L+1}(T_r)\} + T_r E_L(T_r)$$

$$\begin{aligned} \hat{T} &= P_f \hat{T}_f + P_r T_r \\ &= \{1 - E_L(T_r)\} \frac{L}{k} \frac{1 - E_{L+1}(T_r)}{1 - E_L(T_r)} + T_r E_L(T_r) \\ &= \left\{ \frac{L}{k} 1 - E_{L+1}(T_r) \right\} + T_r E_L(T_r) \end{aligned}$$

$$(\text{Prop 7}) \quad T^* = \frac{L}{k}$$

(Prop 8)

$$\frac{d(P_f)}{d(kT_r)} = P_{L-1}(T_r)$$

and

$$\frac{d(P_r)}{d(kT_r)} = -P_{L-1}(T_r)$$

This result follows simply by differentiating the expressions for P_f and P_r of (prop 1) and (prop 2) analytically.

$$\begin{aligned} P_r &= E_L(T_r) \\ &= \sum_{i=0}^{L-1} \frac{(kT_r)^i}{i!} e^{-kT_r} \\ &= e^{-kT_r} + \sum_{i=1}^{L-1} \frac{(kT_r)^i}{i!} e^{-kT_r} \end{aligned}$$

Thus

$$\begin{aligned}
\frac{d(P_r)}{d(kT_r)} &= -e^{-kT_r} + \sum_{i=1}^{L-1} \frac{i(kT_r)^{i-1}}{i!} - \frac{(kT_r)^i}{i!} e^{-kT_r} \\
&= \sum_{i=1}^{L-1} \frac{(kT_r)^{i-1}}{(i-1)!} e^{-kT_r} - \sum_{i=0}^{L-1} \frac{(kT_r)^i}{i!} e^{-kT_r} \\
&= E_{L-1} - E_L \\
&= -P_{L-1}(T_r)
\end{aligned}$$

Moreover since

$$P_f = 1 - P_r$$

$$\frac{d(P_f)}{d(kT_r)} = \frac{d(1-P_r)}{d(kT_r)} = P_{L-1}(T_r)$$

$$(\text{Prop 9}) \quad \frac{d(\hat{kT}_f)}{d(kT_r)} = L \frac{[1 - E_L(T_r)] P_L(T_r) - [1 - E_{L+1}(T_r)] P_{L-1}(T_r)}{[1 - E_L(T_r)]^2}$$

From (prop 3)

$$k\hat{T}_f = L \frac{1 - E_{L+1}(T_r)}{1 - E_L(T_r)}$$

Thus by direct differentiation

$$\frac{d(\hat{kT}_f)}{d(kT_r)} = L \frac{[1 - E_L(T_r)] P_L(T_r) - [1 - E_{L+1}(T_r)] P_{L-1}(T_r)}{[1 - E_L(T_r)]^2}$$

$$(\text{Prop 10}) \quad \frac{d(\hat{kT})}{d(kT_r)} = E_L(T_r)$$

From (prop 6)

$$\hat{T} = \frac{L}{K} [1 - E_{L+1}(T_r)] + T_r E_L(T_r)$$

hence

$$k\hat{T} = L \{1 - E_{L+1}(T_r)\} + kT_r E_L(T_r)$$

Thus by direct differentiation

$$\begin{aligned} \frac{d(k\hat{T})}{d(kT_r)} &= L(p_L(T_r)) + (kT_r(-p_{L-1}(T_r))) + E_L(T_r) \\ &= L p_L(T_r) - kT_r p_{L-1}(T_r) + E_L(T_r) \end{aligned}$$

Since

$$\begin{aligned} L p_L(T_r) &= L \cdot \frac{(kT_r)^L}{L!} e^{-kT_r} \\ &= (kT_r) \frac{(kT_r)^{L-1}}{(L-1)!} e^{-kT_r} \\ &= kT_r p_{L-1}(T_r) \end{aligned}$$

this reduces to

$$\frac{d(k\hat{T})}{d(kT_r)} = E_L(T_r)$$

as required.

Given the above statistics for replacement, on-line failure, and expected time to failure of a component with estimated lifetime, L and assumed replacement time T_r we desire to choose T_r (given L) which minimizes some appropriate cost function. Intuitively, this cost function should represent both the cost of on-line failure and the cost of wasted component lifetime due to replacing components be-

fore failure.^{5,17} We, therefore, adopt the cost functional

$$\text{Cost} = C_f P_f + C_W (kT^* - k\hat{T})$$

Here, C_f and C_W are appropriate weight factors representing the cost of a on-line failure and the cost of component lifetime wastage, respectively. Thus, the first term in the cost functional represents the expected cost of a failure (i.e. the probability of an on-line failure times the cost of such a failure) whereas, the second term in the cost functional represents the expected cost of wasted component lifetime (i.e. the expected lifetime reduction times the cost per unit time for such a lifetime reduction with k serving as a normalizing factor).

To minimize the cost functional of equation 33. one simply substitutes the values for $P_f(T_r)$, T^* , and $\hat{T}(T_r)$ computed in the preceeding pages, defferentiating the Cost with respect to kT_r and setting it equal to zero. This then results in the equality⁹

$$12. \quad 0 = C_f P_{L-1}(T_r) - C_W E_L(T_r)$$

where $d(P_f)/d(kT_r)$ is given by Prop 9 and $d(k\hat{T})/d(kT_r)$ is given by Prop 10. Thus the choice of an optimal T_r given L is reduced to the solution of a single nonlinear equation in one unknown. The solutions of this equation are plotted in Figure 1. for a number of values of L and C_f/C_W . Indeed, it can be readily shown that equation 12. has exactly one solution for $T_r > 0$. Moreover, the function

$$R_L(t) = C_f P_{L-1}(t) - C_W E_L(t)$$

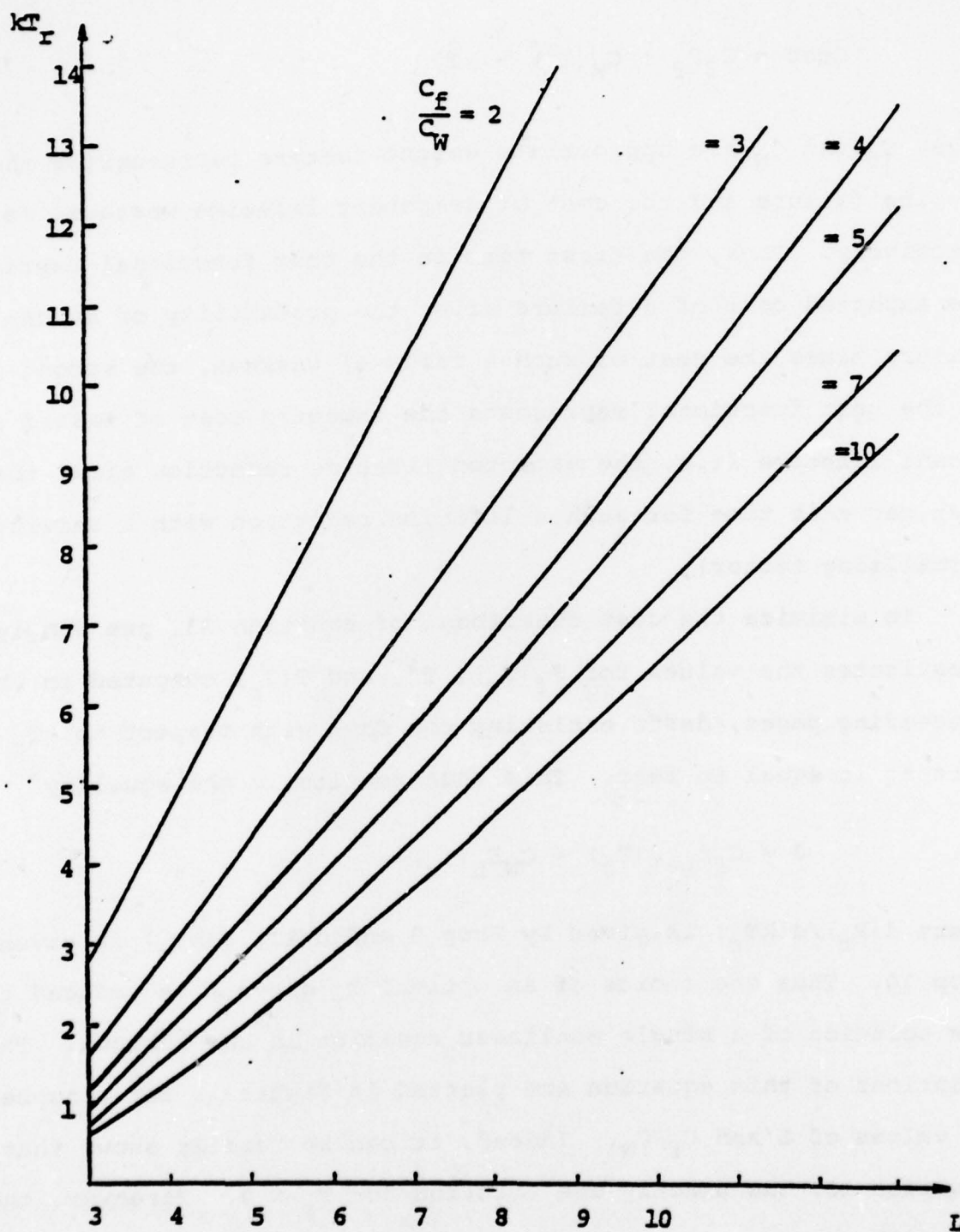


Figure 1. Replacement time (kT_r) v.s. Lifetime L with different weight constant

takes on negative values for $0 < t < T_r$ and positive values for $T_r < t$ hence in an on-line maintenance system one need not even solve equation 12. Rather, one simply evaluates $R_L(t)$ at the time of the next scheduled maintenance. If this results in a negative number, the next scheduled maintenance preceeds the optimal replacement time and hence, we should wait, at least, until the next scheduled maintenance (when we will have more data) to replace the component. On the other hand, if the evaluation of $R_L(t)$ at the next scheduled maintenance time results in a positive value, the optimal replacement time will have passed by the next scheduled maintenance and hence, the component should be replaced at the present maintenance interval.

Summarizing the on-line maintenance algorithm resulting from the above theory takes the following form. At the g th scheduled maintenance interval (at time gT) one measures the component parameter, C_g . If C_g is already out of tolerance, the component is simply replaced and no further analysis is required. If, however, C_g is in tolerance ($C_g > 0$ in our notation) it is used together with the values of the component parameter measured at the previous maintenance intervals to estimate the dynamics of the failure model for the component. Here, sequential refinement schemes may be used both to include the effect of C_g on the estimates made at the $(g-1)$ st maintenance interval and to increase the order of the polynomial used to represent the component failure dynamics. Once the component failure dynamics have been satisfactorily estimated, one solves 6. to estimate whether or not to replace the component. If $R_L((g+1)T) \leq 0$ the component is replaced, whereas, if

$R_L((g+1)T) < 0$ the component is not replaced, and the system is returned to service until the next scheduled maintenance.

V. Simulations

A computer simulation of an on-line periodic maintenance based on the above described algorithm was performed for 600 maintenance intervals with a new component replacing the old component after each replacement decision and/or on-line failure.⁹ The system was subjected to computer generated Poisson shocks with constant $k = 0.1$ shocks per hour and a maintenance interval of $T = 20$ hours. The simulation was first run using identical components with $L = 28$ (expected lifetime of 14 maintenance intervals) and then repeated using random components and noisy data measurements.

For the case where identical components were employed, Table 1. gives the total number of replacements and failures resulting from the application of the algorithm over the 600 simulated maintenance intervals with different values of C_f/C_W . For comparison, Table 2. shows the total number of replacements and failures which would have resulted from a fixed replacement strategy ranging from 6 to 12 maintenance intervals. Since the cost function is

$$\text{Cost} = C_f P_f + C_W (kT^* - k\hat{T})$$

the overall cost can be expressed as

$$\begin{aligned} \text{Cost} = & \frac{C_f}{C_W} (\text{No. of failures}) \\ & + 0.1 (280 * (\text{No. of components used}) - 12000) \end{aligned}$$

The overall costs resulting from the application of our algorithm

and the various fixed replacement schedules may be computed from the data in Tables 1 and 2. The resultant costs for different values of C_f/C_w are given in Table 3.

Note, since $L = 28$ for each component in this simulation, an optimal replacement strategy of approximately 10 maintenance intervals can be computed from equation 12. without estimating L . As such, it is not surprising that this fixed replacement strategy resulted in lower costs than the algorithm. It should, however, be noted that the algorithm did not have the advantage of a a-priori knowledge of L and yet it still out-performed all of the fixed replacement strategies except the optimal strategy (that is, optimal once L is known).

In our second simulation, random noise was added to the data to simulate both the effects of imperfect measurements and the effect components with random failure characteristics. Various simulations were run, as before, for 600 maintenance intervals each with $k = 0.1$ and $T = 20$, with noise levels ranging between 20 and 60 percent of the tolerance interval. The results of these simulations are given in Tables 4. and 5. Except for a single case, which we believe to be anomolous, the algorithm out-performed any fixed replacement strategy.

VI. Conclusion

In the preceeding, we have described a curve fitting algorithm for the prediction of failures in analog devices. The algorithm was tested in a variety of situations and found to be surprisingly effective in predicting failures with relatively little wastage of component lifetime and on-line failure cost.

C_f/C_w	No. of replacement	No. of failure
50	48	7
75	56	1
100	52	2
150	54	2
200	54	2

TABLE 1

Total replacements and failures within 600 maintenance intervals for different C_f/C_w

Constant replacement time	No. of replacement	No. of failure
every 6 intervals	100	0
every 7 intervals	85	0
every 8 intervals	75	0
every 9 intervals	65	1
every 10 intervals	59	1
every 11 intervals	48	6
every 12 intervals	39	11

TABLE 2

Total replacements and failures within 600 maintenance intervals for various fixed replacement strategies

Cost C_f/C_w Methods	50	75	100	150	200
every 6 intervals	1600	1600	1600	1600	1600
every 7 intervals	1096	1096	1096	1096	1096
every 8 intervals	900	900	900	900	900
every 9 intervals	698	723	748	798	848
every 10 intervals	530	555	580	630	680
every 11 intervals	612	762	912	1212	1512
every 12 intervals	750	1025	1300	1850	2400
the algorithm	690	471	512	668	768

TABLE 3

Overall cost with different methods and different C_f/C_w

noise level method	20 %			30 %			40 %			60 %		
	No. of replace fail	No. of replace fail	No. of replace fail	No. of replace fail	No. of replace fail	No. of replace fail	No. of replace fail	No. of replace fail	No. of replace fail	No. of replace fail	No. of replace fail	No. of replace fail
every 6 intervals	100	0	100	0	100	0	100	0	94	6		
every 7 intervals	85	0	85	0	84	1	78	8				
every 8 intervals	75	0	72	3	71	4	64	12				
every 9 intervals	64	2	63	3	60	7	52	17				
every 10 intervals	56	4	51	9	45	15	45	18				
every 11 intervals	45	10	45	10	45	10	39	20				
every 12 intervals	36	15	35	16	36	17	31	23				
the algorithm	56	3	55	5	55	5	50	11				

TABLE 4

Total replacements and failures within 600 maintenance intervals for various fixed replacement strategies and the algorithm at different noise levels.

cost noise level method	0 %	20 %	30 %	40 %	60 %
every 6 intervals	1600	1600	1600	1600	2200
every 7 intervals	1096	1096	1096	1280	2008
every 8 intervals	900	900	1200	1300	2128
every 9 intervals	748	848	948	1376	2432
every 10 intervals	580	880	1380	1980	2364
every 11 intervals	912	1340	1340	1340	2452
every 12 intervals	1300	1728	1828	1984	2612
the algorithm	512	752	980	752	1608

TABLE 5

Overall cost for different methods at different
noise levels

REFERENCES

1. Saeks, R., and S.R. Liberty, Rational Fault Analysis, New York, Marcel Dekker, Inc., 1977.
2. R. Saeks, "An Approach to Built-In Testing", Unpublished Notes, Texas Tech University, Lubbock, Texas, 1977.
3. L. Tung, M.S. Thesis, Texas Tech University, Lubbock, Texas, 1975.
4. Tung, L., and R. Saeks, "An Experiment in Fault Prediction", Proc. of the 4th Inter. Symp. on Reliability in Electronics, Budapest, October 1977, pp. 249-257.
5. Tung, L., Liberty, S.R., and R. Saeks, "Fault Prediction Towards a Mathematical Theory", in Rational Fault Analysis, New York, Marcel Dekker, pp. 135-142, 1977.
6. Esary, J.D., Marshall, A.W., and F. Proschan, Shock Models and Wear Processes, Ann. Probability, Vol. 1, 1973.
7. J.D. Esary, Ph.D. Dissertation, Univ. of California, Berkeley, 1957.
8. A.W. Marshall, "Some Comments on the Hazard Gradient", J. Stoch. Proc. Applic., Vol. 3, 1975.
9. K.S. Lu, Ph.D. Thesis, Texas Tech University, Lubbock, TX, 1977.
10. A. Papoulis, Probability, Random Variable, and Stochastic Processes, McGraw-Hill Book Comp., New York, 1965.
11. Norman I. Johnson, Discrete Distribution, John Wiley and Sons Inc., New York,
12. Rao, C.R., and S.K. Mitra, Generalized Inverse of Matrices and Its Applications, New York, John Wiley and Sons, 1971.
13. K.S. Lu, MSEE Thesis, Texas Tech University, Lubbock, TX, 1973.

FAULT ISOLATION VIA AFFINIZATION*

R. Saeks, R.A. DeCarlo**, and S. Sangani***

*This research supported in part by Office of Naval Research Contracts 75-C-0924 and 76-C-1136.

**Presently with the School of Electrical Engineering, Purdue University.

***Presently with Bell Laboratories Incorporated, Columbus, Ohio.

FAULT ISOLATION VIA AFFINIZATION*

R. Saeks and R. DeCarlo
Dept. of Electrical Engineering
Texas Tech University
Lubbock, Texas 79409

S. Sangani
Dept. of Electrical Engineering
University of Detroit
Detroit, Michigan

Abstract

It is shown that by using an affine (linear plus constant) approximation for the components in a nonlinear analog circuit that it is possible to do fault isolation for a nonlinear circuit via essentially linear techniques.

INTRODUCTION

In this paper we discuss the problem of fault isolation as opposed to fault detection. Briefly however, in the fault detection arena, linearization of components in a circuit works very well. Heuristically, "the linearization of a circuit is faulty if and only if the circuit is faulty." One may concoct artificial counter examples to the necessity of this condition, yet the concept has been successfully utilized in "real world" circuits.

Unfortunately for fault isolation the linearization concept fails in many cases. For example suppose a biasing resistor in a circuit containing a tunnel diode radically changes its value. At a given bias the nonlinear characteristic of the tunnel diode has a particular slope, linearization. With the breakdown of the bias resistor, a linearization (the slope) of the nonlinear characteristic at the bias point of the

*This research supported in part by Office of Naval Research Contracts 75-C-0924 and 76-C-1136.

RESEARCH DATA NOT FILMED
BLACK

tunnel diode is faulty ignoring the fact that it is operating correctly at a different bias. Such linearizations then are "indifferent" to bias changes.

To alleviate this "indifference" we postulate an affine model of components. For linear components the affinization reduces to a linearization. However, for nonlinear components one associates an ordered pair (a,b) where "a" is the slope of the "nonlinear characteristic" at the bias "b." As applied to the above example, the affinization would have shown that the slope-bias pair was correct but that the biasing resistor had broken down.

The remainder of the paper discusses some examples illustrating the idea of this affinization.

Examples

Suppose we have the circuit of Figure 1 which contains the nonlinear tunnel diode whose characteristic $g(\cdot)$ is shown in Figure 2.

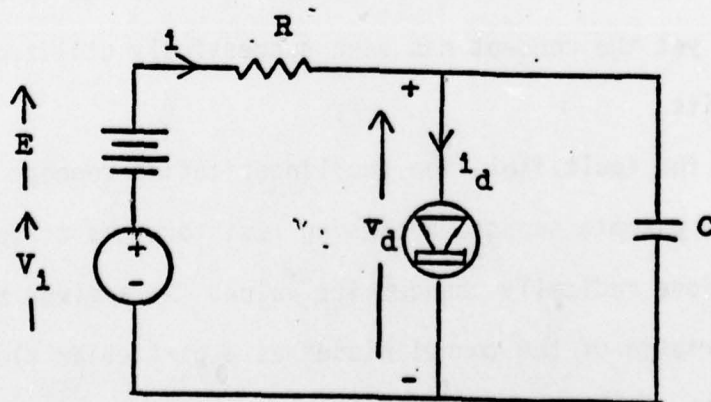


Figure 1. Circuit diagram containing tunnel diode.

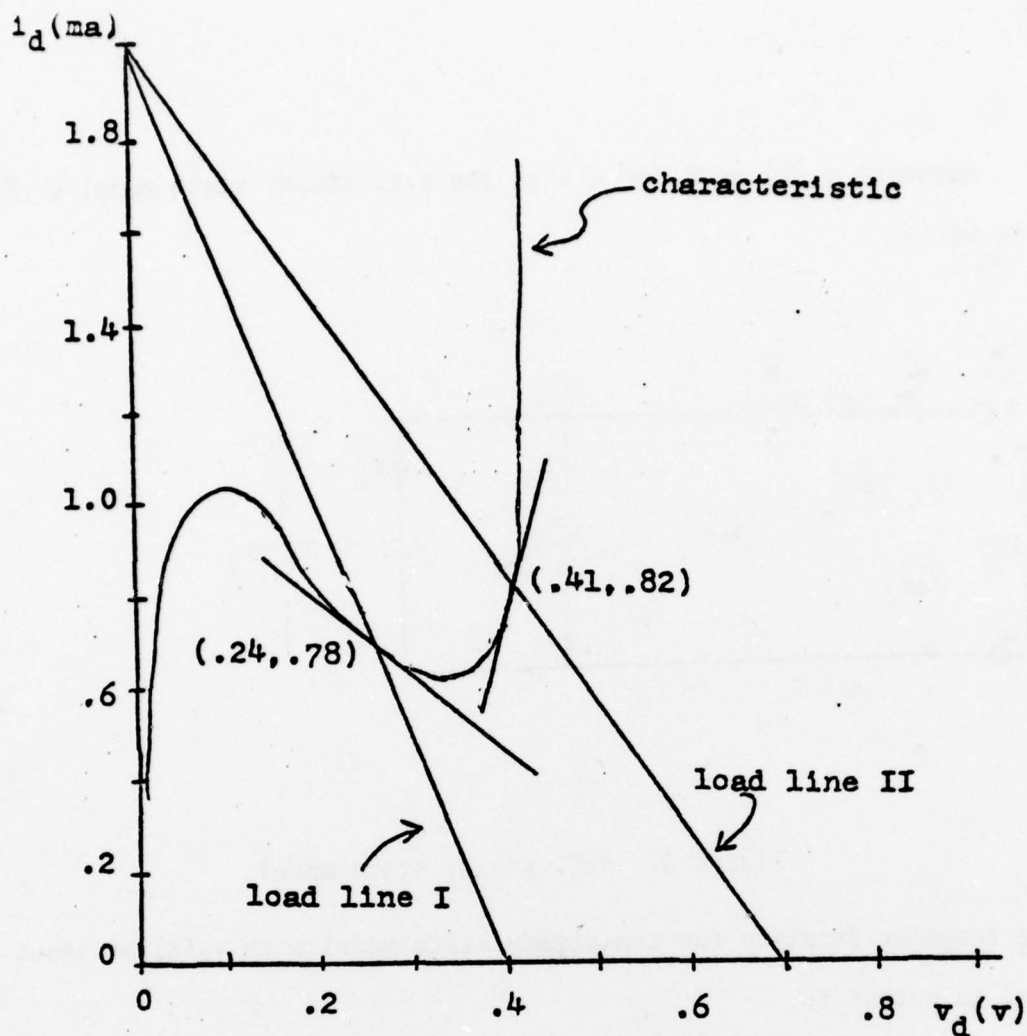


Figure 2. Nonlinear characteristic of tunnel diode.

The Kirchoff laws for this circuit imply

$$(1) \quad E + v_i(t) = Ri(t) + v_c(t)$$

and

$$(2) \quad i_d = g(v_d)$$

Suppose we take the operating point of the tunnel diode to be (V_0, I_0) .

Then assuming $v_i(t) = 0$

$$(3) \quad E = RI_0 = V_0$$

and

$$(4) \quad I_{0d} = g(v_{0d})$$

Assuming $v_i(t) \ll E$ for all t , the a.c. steady state model of Figure 3 is valid.

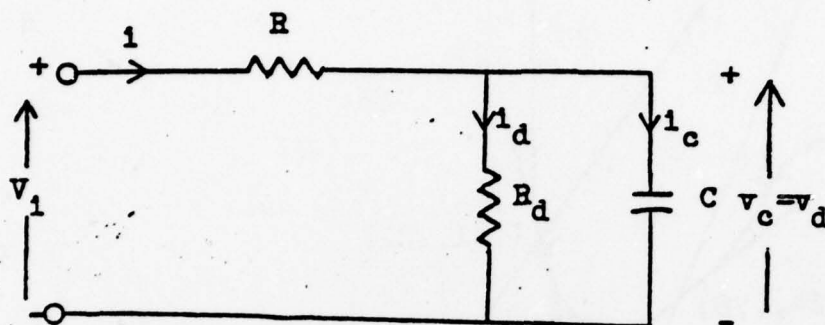


Figure 3. A.C. steady state model.

The transfer function for the steady state model with $v_i(t)$ as input and $i(t)$ as output is

$$(5) \quad H(j\omega) = \frac{1 + j(\omega C R_d)}{(R + R_d) + j(\omega C R_d R)}$$

Example 1: Assume $E = .4$ volts. Suppose the circuit is excited by sin waves of angular frequency $\omega_1 = 10$ rad., $\omega_2 = 5$ rad., and $\omega_3 = 1$ rad. and obtain measurements of $H(j\omega)$ as illustrated in Table 1. Using these measurements

ω	$H(j\omega)$
10 rad	$(4.311 + j2.924) 10^{-3}$
5 rad	$(2.862 + j0.192) 10^{-3}$
1 rad	$(-3.029 + j2.325) 10^{-3}$

TABLE 1

one may solve for $R = 202.7$ ohm, $R_d = -447.0$ ohm and $C = 9.883 \cdot 10^{-4}$ farads. Now stored somewhere in a computer are the true component values

of $R = 200$ ohm, $C = 10^{-3}$ farads, and at the bias point, ($V_0 = .24, I_0 = .78\text{ma}$), the slope of the characteristic of the tunnel diode is $-2.23 \cdot 10^{-3}$ mhos (see Figure 2). Comparing the calculated values of R and C with the true values, one concludes that these components are not faulty. Drawing a load line for $E = .4$ and $R = 202.7$ ohms (load line I of Figure 2) we observe that the bias point is ($V_0 = .24, I_0 = .78$ ma). Now $-1/R_d = 2.237 \cdot 10^{-3}$ mhos which is the correct linearization at the particular bias. This implies the diode is also not faulty. The conclusion is that the circuit works as designed.

Example 2: Here assume $E = .7$ and that the true component values are $R = 200$ ohm, $C = 10^{-3}$ farads, and at the bias point of ($V_0 = .41, I_0 = .82\text{ma}$) (see Figure 2) the linearization (slope) is $3.31 \cdot 10^{-3}$ mhos. Again excite the circuit and take measurements as tabulated in Table 2.

ω	$H(j\omega)$
10 rad	$(2.492 + j2.078) \cdot 10^{-3}$
5 rad	$(2.059 + j1.058) \cdot 10^{-3}$
1 rad	$(1.572 + j2.129) \cdot 10^{-3}$

TABLE 2.

Solving for the component values we obtain $R = 348$ ohm $R_d = 301$ ohm and $C = 9.988 \cdot 10^{-4}$ farads. With $E = .7$ and $R = 348$ ohms draw load line II as in Figure 2. The slope $1/R_d = 3.31 \cdot 10^{-3}$ corresponds to the correct linearized operating characteristic of the tunnel diode. The calculated capacitor value is within tolerance of the true value so it is not faulty. However, the bias resistor is clearly a faulty component. The point here is that if the bias information associated with the tunnel diode were ignored, we would be unable to decide whether the resistor or diode were faulty.

Example 3: Again assume $E = .7$ volts. As with examples 1 and 2, excite the same circuit having the same true component values to obtain data as tabulated in table 3.

ω	$H(j\omega)$
10 rad	$(2.74 + j.372) 10^{-3}$
5 rad	$(2.492 + j.59) 10^{-3}$
1 rad	$(1.66 + j.385) 10^{-3}$

TABLE 3.

Making the required calculations we have $R = 350$ ohms $R_d = 298$ ohms and $C = 1.999 10^{-3}$ farads. The diode operates on load line II of Figure 2 and has the correct linearization. However, both the capacitor and resistor are faulty in this case. Again the point is that without taking into account the bias, one would be unable to decide which component was faulty, although one would be able to conclude that the circuit was faulty.

These examples serve to show that the concept of an affinization is potentially very useful in isolating a component failure. The advantage is that it is only slightly more general than the linearization concept of the fault detection problem.

FAULT ANALYSIS IN AFFINE SEQUENTIAL CIRCUITS*

S. Sangani** and R. Saeks

*This research supported in part by Office of Naval Research Contracts 75-C-0924 and 76-C-1136.

**Presently with Bell Laboratories Inc., Columbus, Ohio.

by

S. Sangani
Dept. of Electrical Engineering
University of Detroit
Detroit, Michigan

and

R. Saeks
Dept. of Electrical Engineering
Texas Tech University
Lubbock, Texas

I. Introduction

The rapid progress in electronic technology has made it necessary for researchers to develop a sound theoretical foundation for fault analysis techniques for electronic circuits and systems which eventually would lead to automated maintenance systems. In recent papers^{1,2,3}, the authors have developed a fault analysis technique for Linear Sequential Circuits (LSCs) based on spectral theoretic approach. This technique parallels the multifrequency testing techniques of analog fault analysis^{4,5,6,7} and by exploiting the circuit dynamics, it requires fewer test inputs for sequential circuits than for a combinational circuits of a similar complexity. The same is true in the analog case where this procedure requires fewer test inputs for a dynamical circuit than for a combinational circuit of a similar complexity. A perfectly valid spectral theory is then developed by viewing an LSC as a linear operator on a sequence space. This spectral theory parallels the steady state frequency domain theory for analog circuits and thus is used to formulate a fault analysis procedure for Linear Sequential Circuits. Initially it is assumed that LSCs fail linearly ("stuck on zero", "open circuit" and "short circuit" but not on "stuck on one" failures) but after which it is extended to the case of Affine Sequential Circuits (ASCs) which fail affinely. This latter generalization permits a slight generalization of

*This research supported in part by ONR contract 75-C-0924

the traditional LSC by allowing NOT gates and bias sources in addition to the usual LSC components. Moreover, it allows the theory to include "stuck on one" faults. In section II, the philosophy of the fault analysis procedure for LSC's is described and references are given for more information. This algorithm is extended to Affine Sequential Circuits and is presented in Section III. Section IV is devoted to family of examples.

II. Fault Analysis in Linear Sequential Circuits^{3,8}.

Mathematically, a Linear Sequential Circuit is characterized by a pair of state equations^{9,10},

$$\begin{aligned} X_{k+1} &= AX_k + BU_k \\ Y_k &= CX_k + JU_k \\ X_0 &= \underline{X} \end{aligned} \quad \infty < k < \infty \quad (1)$$

Where X , U and Y are vectors and A , B , C and J are conformable matrices taking their values in the Galois field $GF(p)$ ¹¹. We interpret these equations as a central value problem. Wherein, we seek to find two-sided sequences; X_k and Y_k , $-\infty < k < \infty$, satisfying equations (1) for a given sequence U_k and a central value $X_0 = \underline{X}$.

Next, we define a "transfer function" in the usual manner via,

$$S(e) = J + C(eI - A)^{-1}B \quad (2)$$

Where e is an element of the algebraic closure of $GF(p)$ ¹¹. $S(e)$ is a matrix of rational functions in e with coefficients in $GF(p)$ whose poles and zeros are well defined elements of the algebraic closure and whose properties are similar to those of classical analog transfer function. The fundamental property of $S(e)$ for the purpose of fault analysis is that for each e (except for the poles of $S(e)$) the sequence, $U_k = e^k$, $-\infty < k < \infty$, is an eigensequence of the difference operator with eigenvalue $S(e)$. The key fact here is that the eigensequence is independent of LSC under study and hence is assured to be eigensequence for a faulty circuit even if the fault is unknown. As such, we test an LSC with various eigensequences from which we compute $S(e)$. The only potential problem arising is that these eigensequences "live" in an extension of the space on which the actual circuit is defined and therefore can not be physically implemented. This problem is solved by obtaining the response of an LSC under study due to an impulsive input and computing $S(e)$ therefrom. See reference^{8,12} for such a formulation. We also compute $S(e)$ in terms of component parameters and connection matrices^{3,8} through the use of the component connection model^{3,7,8}. A set of equations are then obtained, which can be solved for the unknown component parameters. The algorithm for fault analysis in Linear Sequential Circuits is summarized in the following steps.

- (i) Measure the zero state impulse response of an LSC under study.
- (ii) Compute $S(e_i)$ from the impulse response^{8,12} obtained in (i) for various elements e_1, e_2, \dots, e_n in the extension of the finite field over which the LSC under study is defined. Here, the value of n depends on the number of unknown component parameters in the circuit.

*Definition of D-transform - A D-transform is expressed in terms of a formal power series i.e., $D(g(i)) = \sum_{i=-\infty}^{\infty} g(i)D^{-i}$ where $g = I \rightarrow GF(p^m)$ for $\forall g \in G$. Here, G is the set of all sequences and I denotes the natural integers.

- (iii) Obtain $S(e)$ from Component Connection model.
- (iv) Obtain the set of equations for unknown component parameters from step (ii) & (iii) and then solve using the term expansion algorithm^{8,13}.

III. Fault Analysis in Affine Sequential Circuits⁸.

A. Affine Circuits:

Mathematically, a "two-sided" Affine Sequential Circuit over a finite field is represented by a set of difference equation,

$$\begin{aligned} X_{k+1} &= AX_k + BU_k \\ Y_k &= CX_k + JU_k + w_k \quad -\infty < k < \infty \\ X_0 &= \underline{X} \end{aligned} \quad (3)$$

Where $\{w_k\}$ is a sequence of constants and w_k is some constant at time k .

We assume that the circuit components are characterized by the D-transform* equation,

$$b_i(D) = [(g_i)f_i(D)]a_i(D) + (\eta_i(D))(\hat{\psi}_i(D)) \quad i = 1, 2, \dots, n \quad (4)$$

where:

- $a_i(D)$ is the D-transform of the input sequence to the i th component
 - $b_i(D)$ is the D-transform of the output sequence to the i th component
 - $f_i(D)$ is the dynamics of the i th component
 - g_i is a scalar, gain of the linear part of i th component
 - $\hat{\psi}$ represents component dynamics through which bias signal passes.
 - $\eta_i(D)$ is the D-transform of $\eta_i(D)$ from η_i , i.e.,
- $$\eta_i(D) = \left(\frac{D}{D-1}\right)\eta_i \quad (5)$$

Substituting (5) in (4) we get,

$$b_i(D) = [(g_i)f_i(D)]a_i(D) + \left(\frac{D}{D-1}\right)\eta_i\hat{\psi}(D) \quad (6)$$

Letting $\psi_i(D) = \left(\frac{D}{D-1}\right)\hat{\psi}_i(D)$ and since the η_i are scalars, we can write

$$b_i(D) = [(g_i)f_i(D)]a_i(D) + [(\eta_i)\psi_i(D)] \quad (7)$$

For our purposes, it is assumed that the component dynamics $f_i(D)$ remain constant and all faults manifest themselves as changes in g_i and η_i . Such a model includes "stuck-on-one" faults which are not included in the case of LSC's.

One may note that an ASC over GF(2) may be viewed either as an LSC in which constant bias source is added or as an LSC into which NOT gates have been inserted. Both views are equivalent since one can construct a NOT with a biased adder and vice versa.

For notational simplicity, we condense n scalar equations given in (7) into a matrix identity, i.e.,

$$b(D) = [(G)F(D)]a(D) + [n(\psi(D))] \quad (8)$$

The connections are assumed to be characterized by the algebraic constraints^{6,7,8},

$$\begin{bmatrix} a(D) \\ y(D) \end{bmatrix} = \begin{bmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} b(D) \\ u(D) \end{bmatrix} \quad (9)$$

Simultaneous solution of (8) and (9) yields,

$$\begin{aligned} y(D) &= [L_{22} + L_{21}(I - (G)F(D)L_{11})^{-1} \\ &\quad (G)F(D)L_{12}]u(D) + [L_{21}(I - (G)F(D)L_{11})^{-1} \\ &\quad (n)\psi(D)] \end{aligned} \quad (10)$$

Let

$$S(D) = L_{22} + L_{21}(I - (G)F(D)L_{11})^{-1}(G)F(D)L_{12}$$

and

$$T(D) = L_{21}(I - (G)F(D)L_{11})^{-1}(n)\psi(D)$$

Then equation (10) becomes,

$$y(D) = S(D)U(D) + T(D) \quad (11)$$

B. Fault Analysis:

Fault analysis in Affine Sequential Circuits involves developing techniques for determining G and n .

G can be determined by considering the linear part of an ASC (i.e., set $w_k = 0$) and using the procedure outlines in section II. For this purpose, we determine the impulse response of the linear part of an ASC as follows: Observe the response of the ASC under consideration due to $\{U_k\} = \{0\}$. When this response becomes periodic, apply an impulse sequence to this ASC. Then the impulse response of the linear part is given by the difference of two responses i.e., response obtained after and before the impulse is applied. Denote it by $\{h_k\}$.

Determining n :

We compare D-transform of (3) with (11) and get,

$$W(D) = T(D) \text{ where } w(D) = D\{w_k\} \quad (12)$$

i.e., if $\{w_k\}$ is determined from the measurements on an ASC, n can be determined from $T(D)$. As such, w_k is given by⁸ a simple equation, if we assume $X_0 = \underline{X} = 0$.

$$w_k = h_k - h_k \quad (13)$$

Where h_k is the impulse response of an ASC.

With the above theory, we summarize the fault analysis algorithm for ASC in the following steps.

(1) find g_i by using steps (i) through (iv) of

the fault analysis procedure for LSC outlined in section II. "Impulse response of an LSC" should be read as "Impulse response of the linear part of an ASC" in these steps.

- (2) Substitute g_1 's in $T(D)$
- (3) Measure $\{h_k\}$ the impulse response of the ASC under study and obtain $\{w_k\}$.
- (4) Obtain $w(D)$ from $\{w_k\}$.
- (5) Using (12), solve for η , since η is contained in $T(D)$.

IV. Examples

In this section two examples are presented. Both examples illustrate the fault analysis algorithm discussed in section III. The second example also illustrates how stuck-on-one faults are modeled using Affine Sequential Circuits.

Example 1:

Consider the ASC shown in figure 1 defined over $GF(2)$.

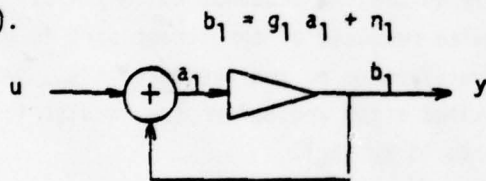


Figure 1

Assume that the impulse response of the linear part of an ASC is measured and is given by $h_0 = 0, h_1 = 1, h_2 = 1, h_3 = 1, \dots$. The impulse response of the Affine Sequential Circuit is measured and is given by $h_0 = 0, h_1 = 0, h_2 = 1, h_3 = 0, h_4 = 1, h_5 = 0, \dots$. It is desired to compute the values of g_1 and η_1 that are compatible with the given data.

Solution:

The connection equations for this circuit are as follows:

$$\begin{bmatrix} a_1 \\ y \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ u \end{bmatrix} \quad (14)$$

and

$$b_1(D) = [g_1/D]a_1(D) + \left(\frac{D}{D-1}\right) [1/D] \eta_1 \quad (15)$$

$$S(D) = [0 + 1(1 - g_1/D \cdot 1)^{-1} g_1/D \cdot 1] \quad (16)$$

$$= (1 - g_1/D)^{-1} g_1/D \quad (16)$$

$$\begin{aligned} T(D) &= \left(\frac{D}{D-1}\right) [1 \cdot (1 - g_1/D \cdot 1)^{-1} (1/D)] \\ &= (1 - g_1/D)^{-1} \left(\frac{1}{D-1}\right) \end{aligned} \quad (17)$$

Calculating g_1 :

Choose an element $e_1 = \alpha$ from $GF(2^2)$ where α is an indeterminate. Then use (2.25) to compute $S(e_1) = S(\alpha)$ from the given impulse response of the linear part of an ASC.

From (2.25) one obtains,

$$S(\alpha)\alpha^1 = b_{-1} + 1 \quad (18)$$

$$S(\alpha)\alpha^2 = b_{-1} + 1 \quad (19)$$

Simultaneous solution of (5.19) and (5.20) yields,

$$S(\alpha) = \alpha. \quad (20)$$

Since $S(e_1) = S(D)_{D=e_1=\alpha} = S(\alpha)$, one can replace D by α in (5.17) and obtain,

$$S(\alpha) = (1 - \frac{g_1}{\alpha})^{-1} \frac{g_1}{\alpha} \quad (21)$$

Simultaneous solution of (5.21) and (5.22) yields

$$g_1 = 1.$$

Substitute $g_1 = 1$ in (5.18) to obtain.

$$\begin{aligned} T(D) &= (1 - \frac{1}{D})^{-1} \left(\frac{1}{D-1}\right) \\ &= \left(\frac{D}{D-1}\right) \left(\frac{1}{D-1}\right) \\ &= \frac{D}{(1+D)^2} \quad [\text{since } -1 = +1 \text{ in } GF(2)] \\ &= \frac{D}{1+D^2} \end{aligned} \quad (22)$$

Obtaining $\{w_k\}$: from (5.14) one obtains,

$$\begin{aligned} w_0 &= h_0 - h'_0 = 0 - 0 = 0 \\ w_1 &= h_1 - h'_1 = 1 - 0 = 1 \\ w_2 &= h_2 - h'_2 = 1 - 1 = 0 \\ w_3 &= h_3 - h'_3 = 1 - 0 = 1 \end{aligned}$$

so, $\{w_k\} = \{0, 1, 0, 1, \dots\}$, such that $w_0 = 0$.

$$\begin{aligned} W(D) &= D\{w_k\} = \{0 \cdot D^0 + 1 \cdot D^{-1} + 0 \cdot D^{-2} \\ &\quad + 1 \cdot D^{-3} + \dots\} \\ W(D) &= \frac{D}{1+D^2} \end{aligned} \quad (23)$$

solving for h:

Substituting (5.23) and (5.24) in (5.10), i.e.,

$$W(D) = T(D)\eta_1$$

$$\text{i.e., } \frac{D}{1+D^2} = \frac{D}{1+D^2} \cdot \eta_1 \quad (24)$$

which yields $\eta_1 = 1$.

It is interesting to note from Figure (1) that b_1 is exactly opposite to a_1 which is a characteristic of the NOT gate. So ASC's also permit modeling of NOT gate for the purpose of fault analysis.

Example 2:

Consider the ASC shown in Figure 5.2. Assume that the impulse response of the linear part of the ASC is measured and is given by $h_0 = 0$, $h_1 = 0$, $h_2 = 0$, The impulse response of the Affine Sequential Circuit is measured and is given by $h'_0 = 0$, $h'_1 = 1$, $h'_2 = 1$, $h'_3 = 1$, It is desired to compute the values of g_1 and η_1 that are compatible with the given data.

The connection equations are the same as those of the previous example. That is, they are given by (5.15), (5.16), (5.17) and (5.18). Calculating g_1 :

Choose an element $e_1 = \alpha$ from $GF(2^2)$. Use (2.25) and the impulse response of the linear part of an ASC to compute $S(\alpha)$. This is given by

$$S(\alpha) = 0. \quad (25)$$

Simultaneous solution of (5.26) and (5.22) yields $g_1 = 0$. Substitute $g_1 = 0$ in (5.18) to obtain,

$$T(D) = \frac{1}{D+1} \quad (26)$$

Obtaining $\{w_k\}$:

$$\begin{aligned} w_0 &= h_0 - h'_0 = 0 \\ w_1 &= h_1 - h'_1 = 1 \\ w_2 &= h_2 - h'_2 = 1 \end{aligned}$$

$\{w_k\} = \{0, 1, 1, 1, \dots\}$ such that $w_0 = 0$,

$$\begin{aligned} W(D) &= D\{w_k\} = \{0 \cdot D^0 + 1 \cdot D^{-1} + 1 \cdot D^{-2} \\ &\quad + 1 \cdot D^{-3} + \dots\} \\ &= \frac{1}{D+1} \end{aligned} \quad (27)$$

Substituting (5.27) and (5.28) in (5.10) yields $\eta_1 = 1$.

The above example is an illustration of a "stuck-on-one" fault.

References

- Sangani, S.H., Liberty, S.R., and R. Saeks, "Fault Analysis in Sequential Circuits", Proc. of the 17th Midwest Symp. on Circuit Theory, Un. of Kansas, September 1973.
- Sangani, S., Saeks, R., and S. Liberty, "Spectral Theory-Linear Sequential Circuits and the Steady State Frequency Domain", Proc. of the 18th Midwest Symp. on Circuit and System Theory, Concordia University, Montreal, Quebec, Canada.
- Sangani S., Saeks, R., and S. Liberty, "A Spectral Theory Approach to Fault Analysis in Linear Sequential Circuit", Journal of the Franklin Institute, (to appear).

Saeks, R., Singh S.P., and R.W. Liu, "Fault Isolation via Component Simulation", IEEE Trans. on Circuit Theory, Vol. CT-19, pp. 634-660.

Ransom, M.N., and R. Saeks, "Fault Isolation with Insufficient Measurements", IEEE Trans. on Circuit Theory, Vol. CT-20.

Ransom, M.N., "A Functional Approach to Large-Scale Dynamical Systems", Proc. of the 10th Allerton Conf. on Circuits and Systems, pp. 48-55, 1972.

Ransom, M.N. and R. Saeks, "The Connection Function-Theory and Application", Int. Jour. of Circuit Theory and Application, Vol. 3, pp. 5-21, 1975.

Sangani, S.H., Ph.D. Thesis, Texas Tech Univ., August, 1975.

Booth, Taylor L., "Sequential Machines and Automata Theory", N.Y., John Wiley and Sons, Inc., 1967.

Hennie, F.C., "Finite-State Models for Logical Machines", N.Y., John Wiley and Sons, Inc., 1967.

Artin, E., "Galois Theory", Univ. of Notre Dame Press, Notre Dame, Indiana, 1942.

Saeks, R., Sangani, S., and S. Liberty, "On the Computation of Rational Z-Transforms (under publication).

Ransom, M.N., and R. Saeks, "Fault Isolation via Term Expansion," Proc. of the 3rd Pittsburgh Symp. on Modelling and Simulation.

A MEASURE OF TESTABILITY AND ITS APPLICATION
TO TEST POINT SELECTION - THEORY*

N. Sen** and R. Saeks

*This research supported in part by Office of Naval Research Contracts 75-C-0924 and 76-C-1136.

**Presently with the Datapoint Corporation, San Antonio, Texas.

A MEASURE OF TESTABILITY AND ITS
APPLICATION TO TEST
POINT SELECTION - THEORY

N. Sen and R. Saeks
Dept. of Electrical Engineering
Texas Tech University
Lubbock, Texas 79409

ABSTRACT

This paper discusses the formulation of a measure of Testability for linear analog electronic circuits and systems. Using component connection model one can derive a set of fault analysis equations for the given system. A multifrequency analysis of the fault analysis equations is employed to obtain a measure of testability defined as δ . This measure is a function only of the system under test, and the available test signals, but not the test algorithm - a basic requirement.

I. INTRODUCTION

Research in the area of fault analysis of systems has progressed along two independent paths. Digital fault analysis techniques have utilized combinational properties of the circuit or system components while analog techniques have relied mainly on the dynamical (memory) properties of the components involved. Bibliographies in the two areas have been compiled by Rault (1,2). A quick glance over the reference reflects an interesting observation. Analog fault analysis, though historically older has been left behind by digital fault analysis and more progress has recently been in the latter area.

One of the causes that can be attributed to such an experience is the rapid advances and need of the industry in the area of digital electronics which has served as a great stimulus for researchers. However, it is believed that automated analog fault analysis has an equally important impact on the maintainability and serviceability

of equipment. Considering the time taken for detection, isolation and repair of a fault (for example in terms of MTTR), rather, it is felt that the development of analog fault analysis techniques deserves even more attention.

The purpose of the present paper is to introduce a measure of testability, δ , for analog circuits, which is easy to compute by multifrequency analysis and gives the extent to which the fault can be isolated. Section II summarizes the component connection model and the applicability of the method to different component problems. Based on the component connection equations, the fault analysis equations are introduced in Section III. Section IV contains the development of the said δ parameter. Finally, a simple but rich enough example to bring out the method is illustrated in Section V.

II. THE COMPONENT CONNECTION MODEL

The classical approach of measuring response of a system in terms of its components in the form of a transfer function becomes

RESEARCH PAGE NOT FILLED
BLACK

prohibitive for large scale systems. A relatively new approach based on separating the components of the system from the interconnection has been developed (3). The chief advantage of such a bifurcation lies in the fact that overall system constraints are broken into components constraints and the interconnection constraints. The component constraints depend on the behavior of the component and the connection constraints are determined in terms of linear algebraic equations (Kirchoff's laws, scalar and adder equations etc.). This in fact simplifies the complexity involved in the analysis.

Figures 1(a) and 1(b) illustrate the development of the component connection model. Here u and y represent the vectors containing the inputs and outputs respectively, of the overall system while a and b are vectors whose elements are the inputs and outputs respectively of each individual components of the system. Thus the hatches and the shadow areas in the figure: 1(a) represent the system components and the interconnections separately and the description of the overall system and that of the components is contained in the following two transformations

$$y = Su \quad (1)$$

and

$$b = Za \quad (2)$$

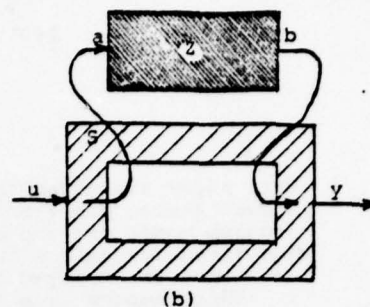
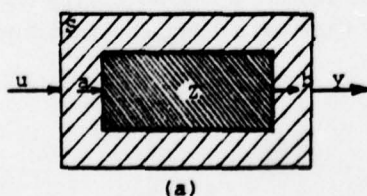


Figure 1. Development of the component connection model.

A separation of the components and connections as shown in figure 1(b) gives us a "donut shape" connection box whose inputs are u and b and outputs are y and a . Their connections are expressible in terms of linear algebraic equations which can further be modeled by the matrix equation:

$$\begin{bmatrix} a \\ y \end{bmatrix} = \begin{bmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} b \\ u \end{bmatrix} \quad (3)$$

Equations (2) and (3) represent the component connection system model. Several studies have been made into the existence and computer implementation of these equations (4-8).

The component equation (2) above contains the operator equation of each component and so when expressed in the frequency domain for a n component system with a sensitivity parameter r included in the component transfer function Z , can also be represented as:

$$\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} = \begin{bmatrix} Z_1(s,r) & & & \\ & Z_2(s,r) & & 0 \\ & & \ddots & \\ & 0 & & Z_n(s,r) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} \quad (4)$$

where each of the $Z(s,r)$'s may represent a different component and so may be:
a capacitor i.e. $Z(s,r) = 1/C(r)s$ where

$C(r)$ is the capacitor

or a delay i.e. $Z(s,r) = f(r)e^{st}$ where $f(r)$ is a function of r and t is the time

or a resistance i.e. $Z(s,r) = R(r)$ where $R(r)$ is the resistance

or an operational amplifier i.e. $Z(s,r) =$

$\frac{K(r)}{(s+p_1(r))(s+p_2(r))}$ where $K(r)$ is the gain

and $p_1(r)$, $p_2(r)$ are poles of the transfer function and so on. Thus a variety of $Z(s,r)$ equations is available and our method is equally applicable in all cases.

III. THE FAULT ANALYSIS EQUATIONS

Using equations (2) and (3) one can obtain the equation:

$$S(s,r) = L_{22} + L_{21} (1 - Z(s,r))^{-1} Z(s,r) L_{12} \quad (5)$$

where s and r are the frequency and the sensitivity parameter respectively.

The above equation relates the overall system operator s to the composite component operator Z . When using the above equation for fault analysis work, the connections are assumed fixed and hence the equation (5) can be considered as a system valued function of a component valued variable $Z(s,r)$. This may be expressed as:

$$S(s,r) = f(Z(s,r)) \quad (6)$$

which is a form applicable to our problem, where all faults manifest themselves in the variations of r with component types and the connections remaining fixed. So the problem is reduced to the measurement of $S(s,r)$ at some frequency and the solution of equation (6) for r . Unfortunately, this solution involves the left invertibility of the matrix:

$$K = L_{12}^T \otimes L_{21} \quad (7)$$

where \otimes denotes the Kronecker matrix product (9). This in turn requires that the system has a large number of test points.

Here we have two alternatives; either add

more test points, or use several frequencies.

Both of them involve increase in cost in terms of hardware or software respectively. Though the former approach is analytically straight and only consists of adding more test points in such a manner that the additional rows corresponding to these test points render K invertible, yet from a practical point of view it is not often possible. The main hurdle is the accessibility of test points. The latter consists in evaluating $S(s,r)$ at several frequencies, for the "same" test points in which case one has to solve the following set of simultaneous equations expressed in the matrix form by:

$$\begin{bmatrix} S(s_1, r) \\ S(s_2, r) \\ S(s_3, r) \\ \vdots \\ S(s_n, r) \end{bmatrix} = \begin{bmatrix} f(Z(s_1, r)) \\ f(Z(s_2, r)) \\ f(Z(s_3, r)) \\ \vdots \\ f(Z(s_n, r)) \end{bmatrix} = F(r) \quad (8)$$

To sum up, the equation solving approach to fault analysis involves yielding more equations in the same number of unknowns (assuming that the component variations with frequency are known and nonfaulty). There is however, a limit to the number of such additional equations based on frequency variation. This is due to the nondynamical components where $F(r)$ is independent of s and the equations obtained for the additional frequencies are redundant. This becomes very important in a case where one is aiming at getting the best out of the available test points but is not aware of this "best". Our theory in the next section gives a method of determining this "best" without actually solving these equations. This will result in a great saving of both the circuit designers and the computer time. When computer implemented, it can serve as an algorithm for test point selection and

hence offering an optimum utilization of various test point options (12).

IV. THE MEASURE OF TESTABILITY

To understand the effect of a small change in a certain parameter (component, temperature or impurity concentration for example in case of solid state circuits) on the whole system, so that it can serve as a basis for fault analysis when approached from the opposite direction, one needs to compute the Jacobian matrix $J_F(\underline{r})$ of $F(\underline{r})$ evaluated at solution of the fault analysis equation, \underline{r} .

A little matrix algebra reveals that the Jacobian of $F(\underline{r})$ evaluated at value of the failed parameter vector \underline{r} , can be expressed as:

$$\frac{\partial f(Z(s, \underline{r}))}{\partial r_i} = L_{21}(1-Z(s, \underline{r})L_{11})^{-1} \frac{\partial Z(s, \underline{r})}{\partial r_i} [I + L_{11}(1-Z(s, \underline{r})L_{11})^{-1}Z(s, \underline{r})]L_{12} \quad (9)$$

where r_i denotes the sensitivity parameter for the i th component.

A use of the following matrix identity (10) has been made in this derivation:

$$\frac{\partial}{\partial x} M^{-1} = -M^{-1} \frac{\partial M}{\partial x} M^{-1} \quad (10)$$

Now if with the available test points and frequencies one can make the Jacobian non-singular (actually a left invertibility will suffice), the system can be completely diagnosed. If this is not so, one can still formulate a test algorithm by using the generalized inverse of the Jacobian, which will locate the solution upto a manifold in the parameter space R^k , whose dimension is equal to the dimension of the null space of the Jacobian. Since, this dimension is an accurate measure of the ambiguity in the test algorithm, it defines a measure of system diagonisibility and minimum possible such dimension of the null space gives the desired measure.

Consistent with the above, we define a local measure of testability at \underline{r} to be:

$$\delta(\underline{r}) = \text{Minimum null } [J_F(\underline{r})] \quad (11)$$

where the minimum is taken over all possible choices of the frequencies $s_1, s_2, s_3, \dots s_n$; $n = 1, 2, \dots$

In a case where only one test frequency is employed it follows from equation (9) that the null space of the Jacobian is equal to the null space of the K matrix given by equation (7) and so with more equations due to additional frequencies one is assured that

$$\delta(\underline{r}) \leq d \quad (12)$$

where d is the dimension of null space of matrix K . Of course, the optimal solution is to have $\delta(\underline{r}) = 0$.

This in other words means that the solution of the nonlinear equation (8) is $\delta(\underline{r})$ dimensional manifold in the neighborhood of \underline{r} . This follows readily from the implicit function theorem. The function involved is invertible if $\delta(\underline{r}) = 0$.

Another important observation concerning the $\delta(\underline{r})$ defined by equation (11) is that it is a property of the equation and is not dependent upon the number of test frequencies chosen.

From above we see that if one is able to compute $\delta(\underline{r})$ accurately, it is possible to get a measure of the uncertainly involved in the solution of equations for Fault Analysis.

There is, however, a problem associated with the computation of $\delta(\underline{r})$. To compute $\delta(\underline{r})$ by equation (11), \underline{r} should be known, which means that the fault is already known and isolated. If this is so, there is no need to compute $\delta(\underline{r})$ since our purpose is to do Fault analysis by $\delta(\underline{r})$.

Hence, there seems to be no other alternative than to adapt the equation solving

approach outlined in Section III. The following theorem, however, proves quite handy in the computation of $\delta(\underline{r})$ by different means.

Theorem 1: $\delta(\underline{r})$ as defined above is equal to the nullity of the set of rational matrices $\left| \frac{\partial f(Z(s, \underline{r}))}{\partial r_i} \right|, i = 1, 2, \dots, n$

viewed as elements of a vector space over the field of complex numbers.

In other words, $\delta(\underline{r})$ is equal to k minus the number of linearly independent

$$\left| \frac{\partial f(Z(s, \underline{r}))}{\partial r_i} \right|, i = 1, 2, \dots, n$$

where k is the dimension of the parameter space R .

The proof for the above theorem which appears in reference (10) brings out an interesting algorithm for choosing test frequencies which no longer have to be picked in advance. For the scalar case, it can be shown that the number of required test frequencies is exactly $k - \delta$ (equal to the rank of the Jacobian). In the general case, however, the number of required test frequencies is less than or equal to $k - \delta$.

Reaching this far, we have achieved a local measure of testability. The problem encountered in computation of a global measure (which necessitates determination of an overall minimum nullity of the Jacobian) is also resolved by the use of the following theorem.

Theorem 2: $\delta(\underline{r})$ is constant "almost everywhere".

Here the term "almost everywhere" is used in the algebraic geometric sense (13), that is, $\delta(\underline{r})$ is constant for all \underline{r} except possibly for the values lying in an algebraic variety of the parameter space R^k .

A proof of this theorem also appears in

reference (10).

Thus this is the key which makes our approach to formulating a measure of testability viable since it allows us to transform the local measure of testability, $\delta(\underline{r})$, into a global measure of testability, δ , which is independent of the failure. In particular, we define δ as the generic value of $\delta(\underline{r})$, i.e. the value taken on by $\delta(\underline{r})$ almost everywhere. As defined, δ is thus independent of the choice of test frequencies employed and the particular failure which one is attempting to diagnose. It is a function only of the system under test, and hence is a measure of testability for the circuit which is independent of both the failure and the test system.

V. AN EXAMPLE

Let us consider the circuit of an amplifier shown in figure 2 below:

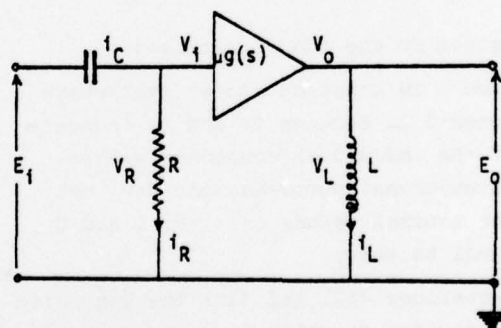


Figure 2. RC-coupled amplifier with an inductive load.

To illustrate the point, we shall assume in the beginning that we have access to E_o , $i_C = i_L$, $i_R = i_C$ and V_i . We can keep all four of them as available outputs in our connection model.

Hence the component equation is given by:

$$\begin{bmatrix} v_o \\ i_L \\ v_C \\ i_R \end{bmatrix} = \begin{bmatrix} u(r)g(s) & 0 & 0 & 0 \\ 0 & 1/L(r)s & 0 & 0 \\ 0 & 0 & 1/C(r)s & 0 \\ 0 & 0 & 0 & 1/R(r) \end{bmatrix} \begin{bmatrix} v_i \\ v_L \\ i_C \\ v_R \end{bmatrix} \quad (13)$$

where r denotes the parameter subject to variation which in our example will be u , R , L and C themselves,

and the connection equation is given by:

$$\begin{bmatrix} v_i \\ v_L \\ i_C \\ v_R \\ \vdots \\ E_o \\ i_o \\ i_R \\ v_i \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_o \\ i_L \\ v_C \\ i_R \\ \vdots \\ E_i \end{bmatrix} \quad (14)$$

which gives us the four L matrices.

Now since s is constant almost everywhere by Theorem 2 in Section IV and is independent of the individual component values, for computational purposes, one can work with the nominal values of u , R , L and C each equal to unity.

Using equations (13) and (14) the composite system transfer function is found with the help of equation (5):

$$S(s, r) = \begin{bmatrix} \frac{s(q(s)+1) + 1}{s+1} \\ \frac{q(s)}{s+1} \\ \frac{s}{s+1} \\ \frac{s}{s+1} \end{bmatrix} \quad (15)$$

For computation of the Jacobian, one can break the equation (9) into:

$$\frac{\partial f(Z(s, r))}{\partial r_i} = L_{21} \frac{\partial Q(Z(s, r))}{\partial r_i} L_{12} \quad (16)$$

where

$$\frac{\partial Q(Z(s, r))}{\partial r_i} = (I - Z(s, r)L_{11})^{-1} \frac{\partial Z(s, r)}{\partial r_i} [L_{11}(I - Z(s, r)L_{11})^{-1} Z(s, r) + I] \quad (17)$$

The equation (17) is computed for each of the internal circuit parameter which is subject to variation (i.e. u , L , C and R) and is evaluated at the nominal value.

$$\frac{\partial Q(Z(s, r))}{\partial u} = \begin{bmatrix} q(s) & 0 & \frac{-q(s)}{s+1} & \frac{-q(s)}{s+1} \\ \frac{q(s)}{s} & 0 & \frac{-q(s)}{s(s+1)} & \frac{-q(s)}{s(s+1)} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (18)$$

$$\frac{\partial Q(Z(s, r))}{\partial L} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{-q(s)}{s} & -1/s & \frac{q(s)}{s(s+1)} & \frac{q(s)}{s(s+1)} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (19)$$

$$\frac{\partial Q(Z(s, r))}{\partial C} = \begin{bmatrix} 0 & 0 & \frac{sq(s)}{(s+1)^2} & \frac{sq(s)}{(s+1)^2} \\ 0 & 0 & \frac{q(s)}{(s+1)^2} & \frac{q(s)}{(s+1)^2} \\ 0 & 0 & -s/(s+1)^2 & \frac{-s}{(s+1)^2} \\ 0 & 0 & \frac{s}{(s+1)^2} & \frac{s}{(s+1)^2} \end{bmatrix} \quad (20)$$

$$\frac{\partial Q(Z(s, \underline{r}))}{\partial C} = \begin{bmatrix} 0 & 0 & \frac{-q(s)}{(s+1)^2} & \frac{sq(s)}{(s+1)^2} \\ 0 & 0 & \frac{-q(s)}{(s+1)^2} & \frac{q(s)}{(s+1)^2} \\ 0 & 0 & 1/(s+1)^2 & -s/(s+1)^2 \\ 0 & 0 & s/(s+1)^2 & -s^2/(s+1)^2 \end{bmatrix} \quad (21)$$

Equations (18) through (21) are further used to compute $\frac{\partial f(Z(s, \underline{r}))}{\partial u}$, $\frac{\partial f(Z(s, \underline{r}))}{\partial L}$,

$\frac{\partial f(Z(s, \underline{r}))}{\partial C}$ and $\frac{\partial f(Z(s, \underline{r}))}{\partial R}$ using equation (16) and the Jacobian is computed by the use of equation (22)

$$J_F(\underline{r}) = \begin{bmatrix} \frac{\partial f(Z(s, \underline{r}))}{\partial u} & \frac{\partial f(Z(s, \underline{r}))}{\partial L} \\ \frac{\partial f(Z(s, \underline{r}))}{\partial C} & \frac{\partial f(Z(s, \underline{r}))}{\partial R} \end{bmatrix} \quad (22)$$

which is found to be:

$$J_F(\underline{r}) = \begin{bmatrix} \frac{sq(s)}{s+1} & 0 & \frac{sq(s)}{(s+1)^2} & \frac{sq(s)}{(s+1)^2} \\ \frac{q(s)}{s+1} & \frac{-q(s)}{s+1} & \frac{q(s)}{(s+1)^2} & \frac{q(s)}{(s+1)^2} \\ 0 & 0 & \frac{s}{(s+1)^2} & \frac{-s}{(s+1)^2} \\ 0 & 0 & \frac{s}{(s+1)^2} & \frac{s}{(s+1)^2} \end{bmatrix} \quad (23)$$

According to the theory outlined in Section IV, we compute:

$$\delta = \delta(\underline{r}) = 4 - \text{Number of linearly independent columns of } J_F(\underline{r}) \quad (24)$$

for various combinations of outputs and hence those of the outputs chosen.

A tabulation of all such combinations is given below:

Case No.	No. of Outputs chosen	Output (s)	δ
1	4	E_o, i_o, i_R, V_i	0
2	3	E_o, i_o, i_R	0
3	3	i_o, i_R, V_i	1
4	3	i_R, V_i, E_o	1
5	3	V_i, E_o, i_o	1
6	2	E_o, i_o	1
7	2	i_o, i_R	1
8	2	i_R, V_i	2
9	2	V_i, E_o	2
10	2	E_o, i_R	1
11	2	i_o, V_i	2
12	1	E_o	2
13	1	i_o	2
14	1	i_R	2
15	1	V_i	3

(25)

A quick glance at the table in (25) gives a measure of testability (and hence its application in fault analysis) for the given circuit. For example in case 1 and in case 2, $\delta = 0$ means that the failure can be isolated exactly, while in cases 3 through 7 and case 10, $\delta = 1$ means that it can be isolated upto an error in one parameter and so on.

REFERENCES

1. Rault, J.C., "A Bibliography on Fault Detection and Location in Digital Systems", Rpt. 74.404 Thomson-CSF Inc., Paris 1974. Also in Rational Fault Analysis (ed. R. Saeks and S.R. Liberty) New York, Marcel Dekker 1977.
2. Rault, J.C., "Fault Detection and Location in Analog Circuits - A Bibliography", IEEE Trans on Circuit Theory, Vol. CT-19, pp. 634-640. Also in Rational Fault Analysis (ed. R. Saeks and S.R. Liberty), New York, Marcel Dekker, 1977.
3. Ransom, M.N. and R. Saeks, "The Connection Function Theory and Application", Inter. Journal of Circuit Theory and Application, Vol. 3, 1975.

4. Singh, S.P. and R.W. Liu, "Existence of State Equation Representation of Linear Large-Scale Dynamical Systems", IEEE Trans. Circuit Theory, CT-20, pp. 239-246, (1973).
5. Singh, S.P., "Structural Properties of Large-Scale Dynamical Systems", PhD. Thesis, Univ. of Notre Dame, 1972.
6. Prasad, N. and J. Reiss, "The Digital Simulation of Interconnected Systems", Proc. Conf. Inter. Assoc. of Cybernetics, Belgium, 1970.
7. Trauboth, H. and N. Prasad, "MARSYAS - A Software System for the Simulation of Digital System", Proc. Spring Joint Comp. Conf., 223-235 (1970).
8. Singh, S.P. and H. Trauboth, "MARSYAS", IEEE Circuits and Systems Society Newsletter, 7 (1973).
9. Neudecker, H., "Some Theorems on Matrix Differentiation with Special Reference to Kronecker Matrix Products", Amer. Stat. Assoc. Journal, Sept. 1969, pp. 953-963.
10. Sen, N., M.S. Thesis, Texas Tech Univ., (in preparation).
11. Ransom, M.N. and R. Saeks, "A Functional Approach to Fault Analysis", in Rational Fault Analysis (ed. R. Saeks and S.R. Liberty), New York, Marcel Dekker, 1977.
12. Sen, N. and R. Saeks, "A Measure of Testability and its Application to Test Point Selection-Computation", Proc. IEEE, AUTOTESTCON, Boston, Nov. 1977 (to appear).
13. Baldassarri, M., Algebraic Varieties, Berlin, Springer Verlag, 1956.
14. Young, D.M. and R.T. Gregory, A Survey of Numerical Mathematics, Vol. I and Vol. II, Mass. Addison Wesley, 1973.

A MEASURE OF TESTABILITY AND ITS APPLICATION TO TEST POINT
SELECTION - COMPUTATION*

N. Sen** and R. Saeks

*This research supported in part by Office of Naval Research Contracts
75-C-0924 and 76-C-1136.

**Presently with the Datapoint Corporation, San Antonio, Texas.

A MEASURE OF TESTABILITY AND ITS APPLICATION TO TEST POINT SELECTION - COMPUTATION

N. Sen and R. Saeks

Department of Electrical Engineering
Texas Tech University
Lubbock, Texas 79409

Abstract

This paper discusses the computation of a measure of testability for linear analog circuits and systems. The theory supporting this measure has been developed in an earlier reference. The present paper reflects on the computational aspects involved and presents examples based thereon.

I. Introduction

There has been considerable interest recently in establishing a measure of testability for electronic circuits and systems. Such a measure may be used by the circuit or systems engineer or as an aid in designing testable circuits. Indeed, with increasing circuit complexity, such a measure is rapidly becoming a necessity. This need is reflected in a recent paper by Greenbaum [1]. Though Greenbaum's paper primarily refers to automated testing for digital systems, the general views expressed are quite in order for the analog fault analysis case as well.

The measure proposed in following pages can be easily determined by computer and serves to quantify circuit testability. Furthermore, it may serve as an aid in partitioning a circuit for ease of testing and/or for the choice of test points.

The purpose of the present paper is to formulate a computational algorithm for the determination of this measure of testability. Section II gives a brief review of the theory behind the development of the proposed measure of testability. This is followed in Section III by a development of, and justification for, our computational algorithm.

II. A Review Of Theoretical Aspects

A detailed development of our measure of testability has been presented in an earlier paper [2]. However, in order to permit a meaningful understanding of the ideas to be brought out in Section III, a brief review of the theory involved is given below.

The starting point of our discussion is the

component connection model for a circuit or system [3]. If u and y represent the vectors of inputs and outputs for the overall circuit or system, while a and b are the vectors of component inputs and outputs respectively their overall system equations and the component equations are characterized by the following transformation:

$$y = Su \quad (1)$$

and

$$b = Za \quad (2)$$

These transformations are related by the connection equations:

$$\begin{bmatrix} a \\ y \end{bmatrix} = \begin{bmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} b \\ u \end{bmatrix} \quad (3)$$

Where L_{ij} are matrices with real entries. Now, upon combining equations (1), (2), and (3), we obtain

$$\begin{aligned} S(s,r) &= L_{22} + L_{21} (I - Z(s,r))^{-1} Z(s,r) L_{12} \\ &= f(Z(s,r)) \end{aligned} \quad (4)$$

Where s and r have been included to represent the complex frequency variable and the vector of potentially variable circuit parameter.

Now we assume that all faults manifest themselves in the variations of r with component types and connections are assumed fixed. A basic approach for a single frequency fault analysis will therefore reduce to the solution of equation (4) for the vector r corresponding to the measured value of $S(s,r)$. This unfortunately requires a large number of test points, which is often impractical.

However, a multifrequency analysis of equation (4) offers us a method of obtaining additional equations with the same number of unknowns:

REPRODUCING PAGE NOT FILMED
BLANK

$$S^H = \begin{bmatrix} S(s_1, r) \\ S(s_2, r) \\ S(s_3, r) \\ \vdots \\ S(s_n, r) \end{bmatrix} = \begin{bmatrix} f(Z(s_1, r)) \\ f(Z(s_2, r)) \\ f(Z(s_3, r)) \\ \vdots \\ f(Z(s_n, r)) \end{bmatrix} = F(r) \quad (5)$$

where $f(Z(s, r))$ is as defined by equation (4).

A small variation in one of the potentially variable component parameters has an effect on $F(r)$ which is expressible in terms of the Jacobian Matrix $J_F(r)$ of $F(r)$ evaluated at r , the solution to equation (5).

The system is thus completely diagonosible if $J_F(r)$ is nonsingular, using the available test points and the test frequencies. That is to say the multifrequency fault analysis equations, have a unique solution in a neighborhood of the actual value of the parameter vector. If $J_F(r)$ is singular, a generalized inverse of the Jacobian matrix can be used to solve the fault analysis equations upto a $\delta(r)$ dimensional manifold in the parameter space where $\delta(r)$ is equal to the dimension of the null space of the Jacobian matrix. This dimension therefore defines a local measure of testability at r via:

$$\delta(r) = \text{Minimum} \left[\text{null } J_F(r) \right] \quad (6)$$

Where the minimum is taken over all possible choices of frequencies $s_1, s_2, s_3, \dots, s_n$, $n = 1, 2, 3, \dots$

The necessity of minimizing over a potentially infinite set of possible test frequencies to compute $\delta(r)$ is alleviated by the following theorem:

Theorem 1: $\delta(r)$ as defined by equation (6) above is equal to the nullity of the set of rational matrices $\left\{ \frac{\partial f(Z(s, r))}{\partial r_i}, i = 1, 2, \dots, n \right\}$

Viewed as elements of a vector space over the field of complex numbers.

Further, this local measure of testability $\delta(r)$ becomes a global measure of testability with the aid of the following theorem:

Theorem 2: $\delta(r)$ is constant "almost everywhere".

Proofs of the above theorems appear in reference [4] wherein "almost everywhere" is defined in the algebraic geometric sense. The generic value of $\delta(r)$ is thus taken as our desired global measure of testability and is denoted by δ . This measure of testability is independent of the choice of test frequencies as well as the failure and is thus a function only of the system under test and the location of its test points.

III. An Algorithm To Evaluate δ

Theorem 1 from section II can be expressed as:

$\delta(r) = k$ - No. of linearly independent

$$\left\{ \frac{\partial f(Z(s, r))}{\partial r_i}, i = 1, 2, \dots, k \right\} \quad (7)$$

Where k is the dimension of the parameter space r .

The matrix $\frac{\partial f(Z(s, r))}{\partial r_i}$ in the above

expression can be computed via the following formula [3]:

$$J_F(r) = \frac{\partial f(Z(s, r))}{\partial r_i} = L_{21}(I - Z(s, r)L_{11})^{-1} \frac{\partial Z(s, r)}{\partial r_i} [I + L_{11}(I - Z(s, r)L_{11})^{-1} Z(s, r)L_{12}] \quad (8)$$

where r_i is the potentially variable circuit parameter.

The dimension of the above matrix given by equation (8) is p by q when p is the number of available output test points and q is the number of available input test points.

It is convenient however, to make $J_F(r)$ a column vector, by stacking the various columns, under the first column. Then the linear independence of the matrices for various r_i can be checked using these column vectors whose dimension is pq . Equivalently, the number of linearly independent $J_F(r)$ matrices is equal to the column rank (over the field of complex numbers) of the matrix:

$$M(s) = \left[\text{vec } \frac{\partial f}{\partial r_1} \mid \text{vec } \frac{\partial f}{\partial r_2} \mid \dots \mid \text{vec } \frac{\partial f}{\partial r_k} \right] \quad (9)$$

Where "vec" denotes the operation of a stacking the column of the $J_F(r)$ matrix into a single column.

The problem of computing the column rank of the rational matrix $M(s)$ can be transformed into an equivalent problem of computing the rank of the complex matrix:

$$M_n = \begin{bmatrix} M(s_1) \\ M(s_2) \\ \dots \\ M(s_n) \end{bmatrix} \quad (10)$$

Where the s_j 's are distinct frequencies.

Now if $M(s)V = 0$ for a complex vector V ,

then,

$$M(s_j)V = 0 \quad V \neq 0 \quad (11)$$

Hence,

$$M_n V = \begin{pmatrix} M(s_1) \\ M(s_2) \\ \dots \\ M(s_n) \end{pmatrix} V = 0 \quad (12)$$

Showing that,

$$\text{rank } M_n \leq \text{col rank } M(s) \quad (13)$$

independently of the number and value of the test frequencies.

Conversely if $M_n V = 0$ for sufficiently large n ($> k$ times $M(s)$) $V = 0$, i.e. if a rational function is zero at sufficiently many frequencies, it is identically zero by the fundamental theorem of Algebra.

As such,

$$\text{col rank } M(s) \leq \text{rank } M_n \quad (14)$$

Therefore, from equations (13) and (14) our computational problem is reduced to the computation of

$$\text{rank } M_n = \text{col rank } M(s) \quad (15)$$

where n is taken to be sufficiently large.

Now the rank of M_n equals the rank of the k by k positive semi definite matrix $P = M_n^T M_n$ which is in turn equal to the number of non zero eigenvalues of this P matrix [7].

A tridiagonalization by Householder's method followed by the computation of a Sturm sequence [5-7] may be used to compute the rank of M_n without actually computing the eigenvalues of P .

This procedure can be outlined in the following steps:

- 1) Formulate the component connection model for the system.
- 2) Choose a range and number of frequencies*
- 3) Formulate the $\frac{\partial(f(Z(s,r))}{\partial r_i}$ matrix of

each test frequency.

*Note that the resultant δ is independent of the choice of test frequencies so long as sufficiently many frequencies are employed. As such, we may employ real frequencies for simplifying the resultant computations.

- 4) Form the M_n matrix and the P matrix.

- 5) Use the Sturm sequence algorithm to compute δ .

For the same circuit or system it is possible to carry out the above analysis with various (or all) possible combinations of test inputs and outputs. It is also possible to give cost values to each of these test inputs and outputs depending upon factors such as ease of accessibility, effect on system and test equipment required, etc. Then one can determine an optimum choice of test points compatible with a desired degree of testability.

The authors have developed a computer package based on the above algorithm. Several examples of various passive and active circuits were run on the package and the δ and cost computed. Two of these examples are presented in the next section as a part of illustration.

Details on the Computer Package appear in reference [4] and a complete description along with a program listing is available in reference [7].

IV. Examples

Example 1. As a first example, consider the single stage NPN RC-Coupled transistor amplifier shown in Figure 1 and its hybrid- π equivalent circuit [10] shown in Figure 2. Assuming four outputs available (for comparison purposes) and a single input, a set of connection equations for the equivalent circuit are given in equations (16) and the corresponding set of component equations are given in (17). Since it is not physically possible to isolate a fault in one of the two parallel resistor while they are still connected, a parallel combination R_a of the two resistors R_a and R_b is considered in the equations, thus giving rise to 12 components. As δ is constant almost everywhere and thus independent of the actual component values, for computational purposes one can take the nominal value of all the 12 components to be unity.

The maximum number of possible cases for various combinations of test inputs and outputs is 15.

As per the procedure outlined in section III, a Sturm sequence solution was used to compute δ in each possible case and a tabulation of the values obtained is given in (19). These include costs assigned (for illustrative purpose) to various inputs and outputs according to (18).

The table in (19) gives the measure of testability δ and the associated cost for each case. For example, for a failure to be isolated exactly we require $\delta=0$, it will then be cheapest to select case 5 (with input 1 and

outputs 1 and 2) at a cost of 4. Further, if an error in a single parameter is acceptable (ie $\delta=1$, case 6 will also yield a cost of 4. However, if a $\delta=2$ is acceptable, one can easily see that both cases 2 (with input 1 and output 2) and 3 (with input 1 and output 3) prove to be equally attractive with a cost of 3 each, etc.

Example 2. The second example illustrates the application of a algorithm to a Band Elimination Passive Filter shown in Figure 3.

The other data used is:

Nominal value of each component	= 1
Number of Components used	= 8
Number of potentially variable parameters	= 3
Number of frequencies used	=15 (20)

Here again, it can be seen from (24) that no single output along with the input can give us a fault isolation capability with no error (ie $\delta=0$). With two outputs the optimum choice is case 6 (input and outputs 1 and 3) with $\delta=0$ and cost of 6. However, if $\delta=1$ is acceptable case 1 (with input 1 and output 1) is most acceptable choice with an associated cost of 3.

References

- Greenbaum J.R., "Computer-Aided Fault Analysis Today, Tomorrow, or Never?", in *Rational Fault Analysis*, (ed. R. Saeks and S.R. Liberty), New York, Marcel Dekker, 1977.
- Sen N. and R. Saeks, "A Measure of Testability and its Application to Test Point Selection - Theory", *Proc. 20th Midwest Symposium on Circuits and Systems*, Lubbock, August 1977.
- Ransom, M.N. and R. Saeks, "The Connection Function - Theory and Application", *Inter. Journal of Circuit Theory and Application*, vol. 3, 1975.
- Sen, N., M.S. Thesis, Texas Tech University, Lubbock, Texas, 79409. 1977.
- Young, D.M., and R.T. Gregory, *A Survey of Numerical Mathematics*, vol. 1 and vol. II, Massachusetts, Addison Wesley, 1973.
- Ortega, J., "The Givens-Householder Method for Symmetric Matrices", in *Mathematical Methods for Digital Computers*, by Ralston, A., and H.S. Wilf, Vol II, New York, Wiley, 1967.
- Wilkinson, J.H., *The Algebraic Eigenvalue Problem*, Oxford, Clarendon Press, 1965.
- Sen, N. and Saeks, R. *ECMP - A Software Package for Computation of Measurability and Cost to Aid Test Point Selection*, Texas Tech Univ., Lubbock, Texas 79409, 1977.
- Gray, P.E., and C.L. Searle, *Electronic Principles - Physical Models and Circuits*, New York, Wiley, 1969.
- Wing, O., *Circuit Theory with Computer Methods*, New York, Holt, Rinehart and Winston Inc., 1972.

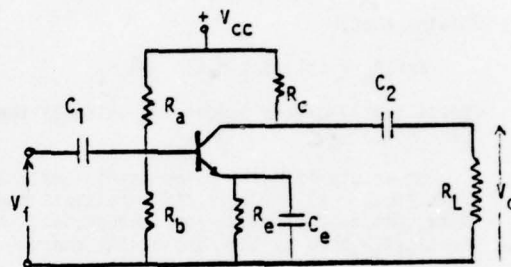


Figure 1. One Stage Transistor Amplifier

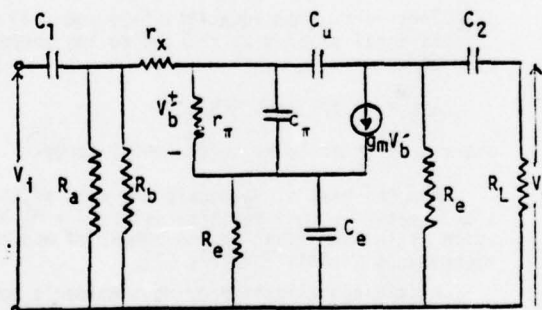


Figure 2. Amplifier Equivalent Circuit

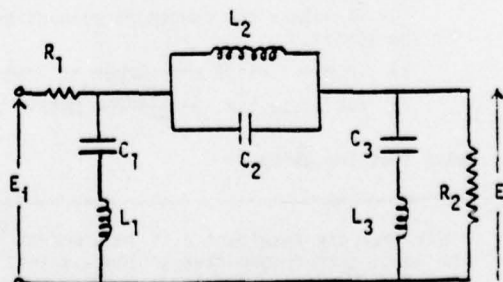


Figure 3. Band Elimination Passive Filter

COMPONENT CONNECTION EQUATIONS (Example 1)

I_{C1}	0	0	0	0	0	1	1	0	1	0	1	1	0	V_{C1}
I_{r_x}	0	0	0	0	0	0	1	0	1	0	1	1	0	V_{r_x}
I_{r_π}	0	0	0	0	0	0	1	-1	1	-1	0	0	0	V_{r_π}
I_{C_u}	0	0	0	0	0	0	0	0	0	1	1	1	0	V_{C_u}
I_{C2}	0	0	0	0	0	0	0	0	0	0	0	1	0	V_{C2}
V_{R_a}	-1	0	0	0	0	0	0	0	0	0	0	0	1	I_{R_a}
V_{R_e}	-1	-1	-1	0	0	0	0	0	0	0	0	0	1	I_{R_e}
V_{C_τ}	0	0	1	0	0	0	0	0	0	0	0	0	0	I_{C_τ}
V_{C_e}	-1	-1	-1	0	0	0	0	0	0	0	0	0	1	I_{C_e}
V_{g_m}	0	0	1	-1	0	0	0	0	0	0	0	0	0	I_{g_m}
V_{R_c}	-1	-1	0	-1	0	0	0	0	0	0	0	0	1	I_{R_c}
V_{R_L}	-1	-1	0	-1	-1	0	0	0	0	0	0	0	1	I_{R_L}
V_o	-1	-1	0	-1	-1	0	0	0	0	0	0	0	1	V_i
I_{C1}	0	0	0	0	0	1	1	0	1	0	1	1	0	
V_{R_a}	-1	0	0	0	0	0	0	0	0	0	0	0	1	
I_e	0	0	0	0	0	0	1	0	1	0	0	0	0	

(16)

V_{C1}	\mathcal{K}_{1s}													I_{C1}
V_{r_x}	r_x													I_{r_x}
V_{r_π}	r_π													I_{r_π}
V_{C_u}	\mathcal{K}_{us}													I_{C_u}
V_{C2}	\mathcal{K}_{2s}													I_{C2}
I_{R_a}	\mathcal{K}_{Ra}													V_{R_a}
I_{R_e}	\mathcal{K}_{Re}													V_{R_e}
I_{C_τ}	$C_\tau s$													V_{C_τ}
I_{C_e}	$C_e s$													V_{C_e}
I_{g_m}	g_m													V_{g_m}
I_{R_c}														V_{R_c}
I_{R_L}														V_{R_L}

(17)

INPUTS AND OUTPUTS (Example 1)

<u>Inputs</u>		
Number Designation	Type Designation	Cost
1	V_1	1

<u>Outputs</u>			(18)
Number Designation	Type Designation	Cost	
1	V_o	1	
2	I_{C1}	2	
3	V_{R_a}	2	
4	I_e	3	

RESULTS (Example 1)

Case	No. of Inputs Chosen	No. of Outputs Chosen	No. Designation of Input chosen	No. Designation of Output Chosen	Cost	Delta
1	1	1	1	1	2	3
2	1	1	1	2	3	2
3	1	1	1	3	3	2
4	1	1	1	4	4	3
5	1	2	1	1,2	4	0
6	1	2	1	1,3	4	1
7	1	2	1	1,4	5	0
8	1	2	1	2,3	5	2
9	1	2	1	2,4	6	1
10	1	2	1	3,4	6	0
11	1	3	1	1,2,3	6	0
12	1	3	1	1,2,4	7	0
13	1	3	1	1,3,4	7	0
14	1	3	1	2,3,4	8	0
15	1	4	1	1,2,3,4	9	0

(19)

COMPONENT CONNECTION EQUATIONS (Example 2)

$$\begin{bmatrix} V_{R1} \\ I_{C1} \\ I_{L1} \\ V_{L2} \\ V_{C2} \\ I_{C3} \\ I_{L3} \\ V_{R2} \\ E_0 \\ I_{C1} \\ V_{C2} \\ I_{C3} \end{bmatrix} = \begin{bmatrix} 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} I_{R1} \\ V_{C1} \\ V_{L1} \\ I_{L2} \\ I_{C2} \\ V_{C3} \\ V_{L3} \\ I_{R2} \\ E_1 \end{bmatrix} \quad (21)$$

$$\begin{bmatrix} I_{R1} \\ V_{C1} \\ V_{L1} \\ I_{L2} \\ I_{C2} \\ V_{C3} \\ V_{L3} \\ I_{R2} \end{bmatrix} \begin{bmatrix} \mathcal{K}_1 & & & & & & & \\ & \mathcal{K}_{C1s} & & & & & & \\ & & \mathcal{K}_{L1s} & & & & & \\ & & & \mathcal{K}_{L2s} & & & & \\ & & & & C_{2s} & & & \\ & & & & & \mathcal{K}_{C3s} & & \\ & & & & & & L_{3s} & \\ & & & & & & & \mathcal{K}_2 \end{bmatrix} \begin{bmatrix} V_{R1} \\ I_{C1} \\ I_{L1} \\ V_{L2} \\ V_{C2} \\ I_{C3} \\ I_{L3} \\ V_{R2} \end{bmatrix} \quad (22)$$

INPUTS AND OUTPUTS (Example 2)

Input

Number Designation	Type Designation	Cost
1	E_i	1

Output

Number Designation	Type Designation	Cost
1	E_o	2
2	I_{C1}	4
3	V_{C2}	3
4	I_{C3}	4

(23)

RESULTS (Example 2)

Case	No. of Inputs Chosen	No. of Outputs Chosen	No. Designation of Inputs Chosen	No. Designation of Outputs Chosen	Cost	Delta
1	1	1	1	1	3	1
2	1	1	1	2	5	2
3	1	1	1	3	4	1
4	1	1	1	4	5	1
5	1	2	1	1,2	7	0
6	1	2	1	1,3	6	0
7	1	2	1	1,4	7	0
8	1	2	1	2,3	8	0
9	1	2	1	2,4	9	0
10	1	2	1	3,4	8	1
11	1	3	1	1,2,3	10	0
12	1	3	1	1,2,4	11	0
13	1	3	1	1,3,4	10	0
14	1	3	1	2,3,4	12	0
15	1	4	1	1,2,3,4	14	0

(24)

A MEASURE OF TESTABILITY AND ITS APPLICATION TO
TEST POINT SELECTION*

N. Sen**

*This research supported in part by Office of Naval Research Contracts
75-C-0924 and 76-C-1136.

**Presently with Datapoint Corporation, San Antonio, Texas.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	ii
LIST OF TABLES	v
LIST OF FIGURES	vi
I. INTRODUCTION	1
1.1 Introduction to Fault Analysis	1
1.2 Fault Analysis in Analog Circuits	2
1.3 The Component Connection Model	3
1.4 The Fault Analysis Equations	7
1.5 The Measure of Testability	9
II. THE MEASURE - THEORETICAL ASPECTS	11
2.1 The Measure - Formulation	11
2.2 The Local Measure - Determination.	13
2.3 The Global Measure	21
2.4 An Analytic Example	26
III. THE MEASURE - COMPUTATIONAL ASPECTS	33
3.1 Computational Approach	33
3.2 The Procedure and Test Point Selection	36
IV. COMPUTER IMPLEMENTATION	38
4.1 Inputs	38
4.2 Outputs	40
4.3 Overall Flowchart	40
4.4 Subroutines	46
V. EXAMPLES	49
5.1 One Stage Transistor Amplifier	49
5.2 Transistor with Load	57

PREVIOUS PAGE NOT FILMED
BLANK

5.3 Band Elimination Passive Filter	62
5.4 Low Pass Active Filter	67
5.5 Wien Bridge Oscillator	75
5.6 Four Component Ladder Network	82
5.7 Six Component Ladder Network	86
5.8 Eight Component Ladder Network	89
5.9 Ten Component Ladder Network	91
5.10 Twelve Component Ladder Network	93
VI. CONCLUSION	95
REFERENCES	97
APPENDIX	100

LIST OF TABLES

Table	Page
2.1 Results (Example 2.1)	32
5.1 Inputs and Outputs (Example 5.1).	54
5.2 Results (Example 5.1)	55
5.3 Inputs and Outputs (Example 5.2).	60
5.4 Results (Example 5.2)	61
5.5 Inputs and Outputs (Example 5.3).	65
5.6 Results (Example 5.3)	66
5.7 Inputs and Outputs (Example 5.4).	71
5.8 Results (Example 5.4)	72
5.9 Inputs and Outputs (Example 5.5).	79
5.10 Results (Example 5.5)	80
5.11 Inputs and Outputs (Example 5.6).	84
5.12 Results (Example 5.6)	85
5.13 Results (Example 5.7)	88
5.14 Results (Example 5.8)	90
5.15 Results (Example 5.9)	92
5.16 Results (Example 5.10).	94

LIST OF FIGURES

Figure	Page
1.1 System Representation	5
1.2 Component-Connection Model Representation	5
2.1 RC Coupled Amplifier with an Inductive Load	27
4.1 The Overall Flowchart	41
4.2 Flow Chart for Computation of Delta	47
5.1 One Stage Transistor Amplifier	51
5.2 Amplifier Equivalent Circuit	51
5.3 Transistor with Load	58
5.4 Transistor Equivalent Circuit	58
5.5 Band Elimination Passive Filter	62
5.6 Low Pass Active Filter	68
5.7 Wien Bridge Oscillator	76
5.8 Oscillator Equivalent Circuit	76
5.9 Four Component RC Ladder Network	82
5.10 Six Component RC Ladder Network	86
5.11 Eight Component RC Ladder Network	89
5.12 Ten Component RC Ladder Network	91
5.13 Twelve Component RC Ladder Network	93

CHAPTER I

INTRODUCTION

1.1 Introduction to Fault Analysis

The rapid advances made in the integrated circuit technology have resulted in the development of new types of digital as well as analog circuits of great complexity. Integration has led to embedding of more and more individual circuit elements in a single chip. Further, a number of such chips along with the interconnection structure may be combined together to construct a large circuit or system. This results in increased complexity of the structure, and hence, more difficult maintenance. If a departure from the anticipated operation of the circuit is observed, the faulty element or elements need to be located. This process of ascertaining whether a circuit or system is operating as expected, is termed as "Fault Detection" while the precise determination of the component or subsystem which failed is called "Fault Isolation" or "Fault Location." "Fault Analysis" concerns itself with techniques to achieve "Fault Detection," "Fault Location," and possibly, "Fault Prediction."

Historically, research in the area of fault analysis of circuits and systems, has progressed along two independent paths. Bibliographies in these two areas have been compiled by Rault, et al. [1, 2]. Digital fault analysis techniques have utilized combinational properties of the circuit or system components, i.e., one tests each component of the circuit by applying a family of test inputs to that circuit. If the expected response is obtained, the circuit is assured of oper-

ating properly and so are all of its components. On the other hand, if the circuit fails to operate correctly for one or more test inputs, the faulty component(s) may be isolated by determining the set of components which are exercised by exactly that set of input test signals. In the case of analog circuits, however, one has the advantage of testing the circuit using signals of different frequencies. Hence, it is possible to measure the gain between various circuit test points at several frequencies. Then, by analytical means, the circuit parameters are determined which yield these gains. If the resulting component parameters are within the specified tolerances, the corresponding component can be assumed to be working, whereas, if a computed component parameter deviates significantly from its operating range, the corresponding component may be assumed faulty.

Fault Analysis techniques for digital and analog circuits with memory differ from each other. While in case of digital circuits [3], the test complexity increases exponentially with memory due to the combinational approach, the opposite is true for analog circuits with memory. Here, one uses multifrequency testing techniques in which one can measure the gain between a pair of test points at several different frequencies simultaneously from a single test signal. This results in a reduction of the total number of test inputs required for a dynamical analog circuit as compared to a memoryless circuit.

1.2 Fault Analysis in Analog Circuits

A survey of references [1] and [2] yields an interesting observation. Analog fault analysis, though historically older, has been left behind by its Digital counterpart and most recent progress has been

in the latter area. One of the sources of this phenomena is the rapid advances and needs of the industry in the area of digital electronics, which has served as a stimulus for researchers. However, it is believed that analog fault analysis has an equally important impact on the maintainability and serviceability of equipment. Considering the time taken for detection, isolation and repair of a fault (for example in terms of MTTR), it is believed that the development of analog fault analysis techniques deserves even more attention. This fact provided a motivation for research in this area and led to the development of the present thesis.

In general, analog fault analysis is heavily predicted on spectral theoretic techniques formulated in a frequency domain setting. In the area of fault analysis of analog circuits, a number of recent papers [4-6] have employed the component connection model of a large scale system.

1.3 The Component Connection Model

The classical approach of modeling a system in terms of its transfer function becomes prohibitive for large scale systems. A relatively new approach based on separating the components of a system from the interconnection has been recently developed [5]. The chief advantage of such a bifurcation lies in the fact that overall system constraints are broken into components constraints and the interconnection constraints. The component constraints depend on the behavior of the decoupled components while the connection constraints take the form of linear algebraic equations (Kirchoff's Laws, scalar and adder equations, etc.). This fact simplifies the analysis.

Figures 1.1 and 1.2 illustrate the development of the component connection model. Here u and y represent the vectors containing the inputs and outputs respectively of the overall system while a and b are vectors whose elements are the inputs and outputs respectively of each individual component. Thus the hatched and shadowed areas in the figure 1.1 represent the system components and the interconnections respectively; and descriptions of the overall system and of the components are given by the following two transformations:

$$y = Su \quad (1.1)$$

and

$$b = Za \quad (1.2)$$

A separation of the components and connections as shown in figure 1.2 gives us a "donut shape" connection box whose inputs are u and b and outputs are y and a . Their connections are expressible in terms of linear algebraic equations which can be modeled by the matrix equation:

$$\begin{bmatrix} a \\ y \end{bmatrix} = \begin{bmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} b \\ u \end{bmatrix} \quad (1.3)$$

Equations (1.2) and (1.3) represent the component connection model for the system. Several studies have been made into the existence and computer implementations of these equations [7-11].

The component connection equation (1.2) above, contains the equation of each decoupled component and so when expressed in the fre-

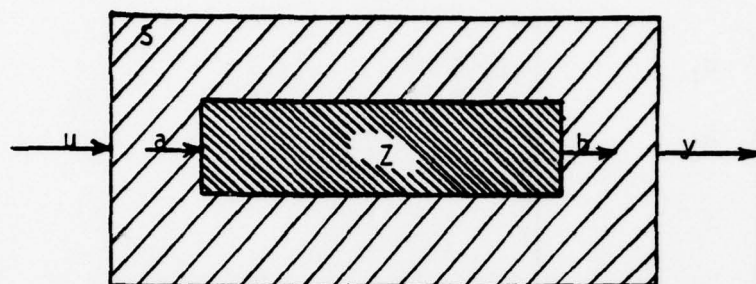


Figure 1.1 System Representation

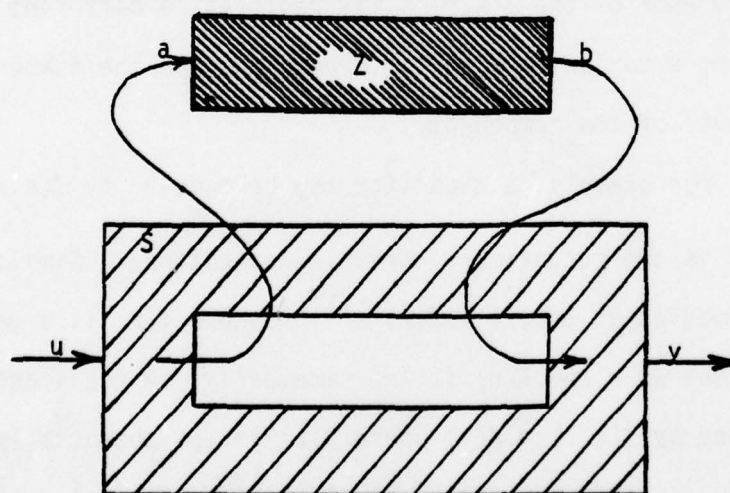


Figure 1.2 Component Connection Model Representation

quency domain for a n component system with a potentially variable parameter vector r included in the component transfer function Z , can also be represented as:

$$\begin{bmatrix} b_1 \\ b_2 \\ \cdot \\ \cdot \\ \cdot \\ b_n \end{bmatrix} = \begin{bmatrix} Z_1(s,r) & & & \\ & Z_2(s,r) & & \\ & & \cdot & \\ & & & \cdot \\ & & & & Z_n(s,r) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \cdot \\ \cdot \\ \cdot \\ a_n \end{bmatrix} \quad (1.4)$$

where each of the $Z(s,r)$'s may represent a different component which may be a scalar or a vector depending upon the number of inputs and outputs of the component.

For example, a capacitor may be modeled by $Z(s,r) = \frac{1}{C(r)}s$ where $C(r)$ is the potentially variable capacitance. Similarly a delay may be modeled as $Z(s,r) = f(r) e^{sT(r)}$ where $f(r)$ is a gain function dependent on r and $T(r)$ is the time delay. A resistance can be represented by $Z(s,r) = R(r)$ where $R(r)$ is the potentially variable resistance. As a more complex example, an operational amplifier can be modeled as $Z(s,r) = \frac{K}{(s + p_1)(s + p_2)}$ where K is the gain and p_1 and p_2 are the poles of the transfer function. Here, the potentially variable circuit parameter vector r can be expressed as Col. (K, p_1, p_2) .

1.4 The Fault Analysis Equations

Using equations (1.2) and (1.3) one can obtain the following equation:

$$S(s,r) = L_{22} + L_{21} (I - Z(s,r) L_{11})^{-1} Z(s,r) L_{12} \quad (1.5)$$

where s and r are the frequency and the potentially variable circuit parameter respectively.

The above equation relates the overall system operator S to the component operator Z . When using this equation for fault analysis work, the connections are assumed fixed and hence the equation (1.5) can be considered as a system valued function of a component valued variable. This may be expressed as:

$$S(s,r) = f(Z(s,r)) \quad (1.6)$$

which is a form applicable to our problem, if it is assumed that all faults manifest themselves in the variations of r with component types and the connections remaining fixed. As such, the fault isolation problem is reduced to the measurement of $S(s,r)$ at some frequency and the solution of equation (1.6) for r . Unfortunately, the solution requires the left invertibility of the matrix [4,11]

$$K = L_{12}^T \otimes L_{21} \quad (1.7)$$

where T denotes the transpose and \otimes indicates the Kronecker matrix product [12]. This in turn requires that the system have a large number of test points.

Here, one has two alternatives; either add more test points, or use several frequencies. Both involve an increase in cost in terms of hardware or software, respectively. The former approach is analytically straightforward and only requires the addition of more test points in such a manner that the additional rows of K corresponding to these new test points render K invertible. From a practical point of view, however, it is often not feasible. The main hurdle is the accessibility of test points. The alternative approach consists of evaluating $S(s,r)$ at several frequencies, for the "same" test points in which case one has to solve the following set of simultaneous equations expressed in the matrix form:

$$\begin{bmatrix} S(s_1, r) \\ S(s_2, r) \\ S(s_3, r) \\ \cdot \\ \cdot \\ \cdot \\ S(s_n, r) \end{bmatrix} = \begin{bmatrix} f(Z(s_1, r)) \\ f(Z(s_2, r)) \\ f(Z(s_3, r)) \\ \cdot \\ \cdot \\ \cdot \\ f(Z(s_n, r)) \end{bmatrix} = F(r) \quad (1.8)$$

To sum up, the multifrequency approach to fault isolation increases the number of equations in the same number of unknowns, (assuming that the component variations with frequency are known and nonfaulty). There is, however, a limit to the number of such additional frequencies which can be employed. This is due to the finite degree

(in s) of the components which renders the equations obtained for the additional frequencies redundant. This becomes significant if one desires to use a minimum number of test points. The theory in Chapter II gives a method of determining maximum number of non-redundant test frequencies without actually solving the equations.

1.5 The Measure of Testability

There has been considerable interest recently, in determining measures of testability for Electronic Circuits and Systems. Such a measure gives a quantitative description of the extent to which faults can be isolated in the circuit or the system. This becomes important as the complexity of the circuit increases, which in turn demands a more formalized theory of testing. The measure proposed in the following chapters is easy to compute by multifrequency analysis and serves to quantify the testability of the circuit. This can be employed in the partitioning of the circuit and the optimum choice of test points at the design stage.

Chapter II presents the basic theory behind the development of our measure of testability with a simple analytic example. Chapter III describes the computational aspects of our measure of testability and the analytic justification for our computational algorithm. This is followed by Chapter IV, where flow charts for the computer implementation of the algorithm for evaluation of the measure are presented. These are based on a computer package ECMP (Electronic Circuit Measurability Package), which is the specific implementation in FORTRAN. Chapter V describes a number of examples run on the

above package. Finally, Chapter VI comments on the conclusions and recommendations for future work.

CHAPTER II

THE MEASURE - THEORETICAL ASPECTS

2.1 The Measure - Formulation

To understand the effect of a small change in a certain circuit parameter (eg: a component, temperature or impurity concentration in case of solid state circuits, etc) on the whole system, so that it can serve as a basis for fault analysis when approached from the opposite direction, one needs to compute the Jacobian matrix $J_f(r)$ evaluated at the solution \underline{r} of the fault analysis equations given by equation (1.8).

A little matrix algebra reveals that the Jacobian matrix of $F(r)$ evaluated at value of the failed parameter vector \underline{r} can be expressed as [5]:

$$J_f(\underline{r}) = \frac{\partial f(Z(s, \underline{r}))}{\partial r_i} = L_{21} (I - Z(s, \underline{r}) L_{11})^{-1} \frac{\partial Z(s, \underline{r})}{\partial r_i} [I + L_{11} (I - Z(s, \underline{r}) L_{11})^{-1} Z(s, \underline{r}) L_{12}] \quad (2.1)$$

where r_i is the potentially variable circuit parameter.

Application of the following matrix identity has been made in the derivation of the above equation:

$$\frac{\partial}{\partial x} M^{-1} = -M^{-1} \frac{\partial M}{\partial x} M^{-1} \quad (2.2)$$

Now by the inverse function theorem [21] if, with available test points and frequencies, one can make the Jacobian matrix $J_f(\underline{r})$ non-

singular (actually a left invertibility will suffice), the system can be completely diagnosed. That is to say the multifrequency fault analysis equations, have a unique solution in a neighborhood of the actual value of the parameter vector. However, if $J_f(\underline{r})$ is singular, a generalized inverse of the Jacobian matrix can be used to "solve" the fault analysis equations up to a manifold in the parameter space whose dimension is equal to the dimension of the null space of the Jacobian matrix. Since this dimension is an accurate measure of the ambiguity in the test algorithm, it defines a measure of system diagnosability and the minimum possible such dimension of the null space gives the desired measure of testability.

Consistent with the above, we define a local measure of testability at \underline{r} to be:

$$\delta(\underline{r}) = \text{Minimum} [\text{null } J_f(\underline{r})] \quad (2.3)$$

where the minimum is taken over all possible choices of frequencies $s_1, s_2, s_3, \dots, s_n; n = 1, 2, 3, \dots$

In a case where only one test frequency is employed, it follows from equation (2.1), that the null space of the Jacobian is equal to the null space of the K matrix given by equation (1.7) and so with more equations due to additional test frequencies one is assured that:

$$\delta(\underline{r}) \leq \tilde{d}$$

where \tilde{d} is the dimension of null space of matrix K. Of course, the

optimal solution is to have $\delta(\underline{r}) = 0$. This, in other words, means that the solution of the non-linear equation (1.8) is a $\delta(\underline{r})$ dimensional manifold in the neighborhood of \underline{r} . This follows readily from the implicit function theorem [21]. The function involved is completely invertible locally if $\delta(\underline{r}) = 0$.

From above, we see that if one is able to compute $\delta(\underline{r})$ accurately, it is possible to get a measure of uncertainty involved in the solution of equations for fault analysis.

2.2 The Local Measure - Determination

Though a definition of local measure $\delta(\underline{r})$ has been given by equation (2.3), it is not possible, so far, to compute $\delta(\underline{r})$ since it involves the determination of "minimum null" where minimum needs to be taken over all possible choices of complex frequencies $s_1, s_2, \dots s_n$; $n = 1, 2, \dots$.

In other words, the determination of $\delta(\underline{r})$ requires minimization over a potentially infinite set of possible test frequencies. This difficulty, is however, alleviated by the use of the following theorem:

Theorem 1: $\delta(\underline{r})$ as defined by equation (2.3) is equal to the nullity

of the set of rational matrices $\left[\frac{\partial f(Z(s, \underline{r}))}{\partial r_i}, i = 1, 2, \dots k \right]$

viewed as elements of a vector space over the field of complex numbers.

Proof:

If S^m is the measured value of the composite system transfer function $S(s, r)$ given by equation (1.6), then another statement of the theorem would be to show that:

$\delta(\underline{r}) = k$ - Number of linearly independent

$$\left[\frac{\partial S^m}{\partial r_i} ; i = 1, 2, \dots k \right] \quad (2.5)$$

where k is the dimension of the parameter vector.

Without loss of generality, we may assume that S^m is a column vector of length d (since, otherwise, it is possible to transform it into a column vector by the "vec" operation [20]). Then $\frac{\partial S^m}{\partial r_i}$ will also

be a column vector of length d for each r_i , $i = 1, 2, \dots k$.

Hence, the proof of the present theorem reduces to the verification of the fact that the number of linearly independent columns in the rational matrix

$$R_d(s) = \begin{bmatrix} R_{11}(s) & R_{12}(s) & . & . & . & R_{1k}(s) \\ R_{21}(s) & R_{22}(s) & . & . & . & R_{2k}(s) \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ R_{d1}(s) & R_{d2}(s) & . & . & . & R_{dk}(s) \end{bmatrix} \quad (2.6)$$

is equal to the maximum possible rank of the following complex matrix:

$$\begin{bmatrix} R_d(s_1) \\ \vdots \\ R_d(s_2) \\ \vdots \\ R_d(s_n) \end{bmatrix} = \begin{bmatrix} R_{11}(s_1) & R_{12}(s_1) & \cdot & \cdot & \cdot & R_{1k}(s_1) \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ R_{d1}(s_1) & R_{d2}(s_1) & \cdot & \cdot & \cdot & R_{dk}(s_1) \\ \hline R_{11}(s_2) & R_{12}(s_2) & \cdot & \cdot & \cdot & R_{1k}(s_2) \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ R_{d1}(s_2) & R_{d2}(s_2) & \cdot & \cdot & \cdot & R_{dk}(s_2) \\ \hline \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \hline R_{11}(s_n) & R_{12}(s_n) & \cdot & \cdot & \cdot & R_{1k}(s_n) \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ R_{d1}(s_n) & R_{d2}(s_n) & \cdot & \cdot & \cdot & R_{dk}(s_n) \end{bmatrix} \quad (2.7)$$

where s_j 's represent all possible choices of complex frequencies s_1 , s_2 , s_3 , ... s_n ; $n = 1, 2, \dots$

Now clearly, if some column of $R(s)$, say the k_{th} is dependent on the remaining columns, then

$$R_{ki}(s) = \sum_{j=1}^{k-1} \alpha_j R_{ji}(s) \quad i = 1, 2, \dots, d \text{ for all } s. \quad (2.8)$$

where the α 's represent constant complex coefficients.

As such, by applying equation (2.8) individually for each frequency, (say f),

$$\text{Col. } R_{ki}(s_f) = \sum_{j=1}^{k-1} \alpha_j \text{Col. } R_{ji}(s_f) \quad i = 1, 2, \dots, d \quad (2.9)$$

for any possible number of choices of s_f .

Therefore, the rank of the matrix of equation (2.7) is less than or equal to the number of linearly independent columns of $R_d(s)$ over the field of complex numbers.

Now, in order to prove that this equality of rank can be achieved with an appropriate choice of complex test frequencies s_f , we come back to consideration of equations (2.6) and (2.7).

Without loss of generality we assume that Col. $[R_{p1}(s) ; p = 1, 2, \dots, d]$ though Col. $[R_{pq}(s) ; p = 1, 2, \dots, d]$ are linearly independent over the field of complex numbers in which case we must show that there exists complex frequencies s_1, s_2, \dots, s_n which make the first q columns of the matrix of equation (2.7) linearly independent ($q \leq k$). If $q = 1$, Col. $[R_{p1}(s) ; p = 1, 2, \dots, d]$ is not identically zero (since otherwise it would not be linearly independent) and hence there exists an s_1 , such that:

$$\text{Col. } [R_{p1}(s_1) ; p = 1, 2, \dots, d] \neq 0 \quad (2.10)$$

As such, the columns in this small $d \times 1$ matrix are linearly independent. With this obvious fact as a starting point, one can use an inductive argument to show that the theorem holds for all values of q . Hence, it can be assumed that it has been shown that for $q = \ell$, there exist m complex frequencies s_1, s_2, \dots, s_m , such that the matrix

$$R_{(md) \times \ell} = \begin{bmatrix} R_{11}(s_1) & R_{12}(s_1) & \cdot & \cdot & \cdot & R_{1\ell}(s_1) \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ R_{d1}(s_1) & R_{d2}(s_1) & \cdot & \cdot & \cdot & R_{d\ell}(s_1) \\ \hline R_{11}(s_2) & R_{12}(s_2) & \cdot & \cdot & \cdot & R_{1\ell}(s_2) \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ R_{d1}(s_2) & R_{d2}(s_2) & \cdot & \cdot & \cdot & R_{d\ell}(s_2) \\ \hline \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \hline R_{11}(s_m) & R_{12}(s_m) & \cdot & \cdot & \cdot & R_{1\ell}(s_m) \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ R_{d1}(s_m) & R_{d2}(s_m) & \cdot & \cdot & \cdot & R_{d\ell}(s_m) \end{bmatrix} \quad (2.11)$$

has ℓ linearly independent columns and we desire to show that there exists an s_{m+1} such that the matrix given by equation (2.12) has $\ell + 1$ linearly independent columns.

$$\begin{array}{c}
 R_{(m+1)d} \times (\ell + 1) = \left[\begin{array}{ccccc}
 R_{11}(s_1) & R_{12}(s_1) & \dots & R_{1\ell}(s_1) & R_{1(\ell+1)}(s_1) \\
 \vdots & \vdots & \ddots & \vdots & \vdots \\
 R_{d1}(s_1) & R_{d2}(s_1) & \dots & R_{d\ell}(s_1) & R_{d(\ell+1)}(s_1) \\
 \hline
 R_{11}(s_2) & R_{12}(s_2) & \dots & R_{1\ell}(s_2) & R_{1(\ell+1)}(s_2) \\
 \vdots & \vdots & \ddots & \vdots & \vdots \\
 R_{d1}(s_2) & R_{d2}(s_2) & \dots & R_{d\ell}(s_2) & R_{d(\ell+1)}(s_2) \\
 \hline
 \vdots & \vdots & \ddots & \vdots & \vdots \\
 \vdots & \vdots & \ddots & \vdots & \vdots \\
 \vdots & \vdots & \ddots & \vdots & \vdots \\
 \hline
 R_{11}(s_m) & R_{12}(s_m) & \dots & R_{1\ell}(s_m) & R_{1(\ell+1)}(s_m) \\
 \vdots & \vdots & \ddots & \vdots & \vdots \\
 R_{d1}(s_m) & R_{d2}(s_m) & \dots & R_{d\ell}(s_m) & R_{d(\ell+1)}(s_m) \\
 \hline
 R_{11}(s_{m+1}) & R_{12}(s_{m+1}) & \dots & R_{1\ell}(s_{m+1}) & R_{1(\ell+1)}(s_{m+1}) \\
 \vdots & \vdots & \ddots & \vdots & \vdots \\
 R_{d1}(s_{m+1}) & R_{d2}(s_{m+1}) & \dots & R_{d\ell}(s_{m+1}) & R_{d(\ell+1)}(s_{m+1})
 \end{array} \right]
 \end{array} \quad (2.12)$$

Since the matrix $R_{(m \times d) \times \ell}$ of equation (2.11) has ℓ linearly independent columns it must contain an ℓ by ℓ non singular submatrix $R_{\ell \times \ell}$ obtained by taking ℓ linearly independent rows of $R_{(m \times d) \times \ell}$.

Now consider the $(\ell + 1)$ by $(\ell + 1)$ submatrix of $R_{[m+1)d] \times (\ell + 1)}$ of equation (2.12) obtained by taking ℓ rows of $R_{[m+1)d] \times (\ell + 1)}$ corresponding to $R_{\ell \times \ell}$ and any single row:

$$[R_{p1}(s_m + 1) \quad R_{p2}(s_m + 1) \quad \dots \quad R_{p(m+1)}(s_m + 1)] \quad (2.13)$$

out of the d rows corresponding to the new frequency $s_m + 1$

Hence,

$$\begin{array}{c} \tilde{R}_{(\ell+1) \times (\ell+1)} = \\ \left[\begin{array}{c|c} \tilde{R}_{\ell \times \ell} & X \\ \hline R_{p1}(s_m + 1) \quad R_{p2}(s_m + 1) \quad R_{p\ell}(s_m + 1) & R_{p(\ell+1)}(s_m + 1) \end{array} \right] \end{array} \quad (2.14)$$

where X depends on the choice of rows in $\tilde{R}_{\ell \times \ell}$. Now we expand

$\tilde{R}_{(\ell+1) \times (\ell+1)}$ by the last row obtaining

$$\begin{aligned} \det. \tilde{R}_{(\ell+1) \times (\ell+1)} = \\ \sum_{j=1}^{\ell+1} (-1)^{\ell+1+j} \times \Delta(\ell+1), j \times R_{pj}(s_m + 1) \end{aligned} \quad (2.15)$$

where $\Delta(\ell+1), j$ denotes the determinant obtained by deleting the $(\ell+1)$ st row and j th column of equation (2.14). Now, since $\tilde{R}_{\ell \times \ell}$ has linearly independent columns:

$$\Delta(\ell+1), (\ell+1) \neq 0 \quad (2.16)$$

Hence, the coefficients in the summation of equation (2.15) are not all zero. As such, the function of equation (2.14) cannot be identically

zero for all $p = 1, 2, 3, \dots, d$, since if it were true, it will imply:

$$\sum_{j=1}^{\ell+1} (-1)^{\ell+1+j} \Delta_{(\ell+1), j} \text{ Col. } [R_{pj}(s_{m+1}); p = 1, 2, \dots, d] \equiv 0 \quad (2.17)$$

and this in turn will contradict the assumed linear independence of

$$\text{Col } [R_{pj}(s) ; p = 1, 2, \dots, d] j = 1, 2, \dots, (\ell + 1) \quad (2.18)$$

over the complex numbers.

Thus there exists a p and a complex number s_{m+1} such that the summation of equation (2.15) is non zero. Using this p and s_{m+1} , $\tilde{R}_{(\ell+1) \times (\ell+1)}$ is a non-singular $(\ell+1)$ by $(\ell+1)$ matrix. However, $\tilde{R}_{(\ell+1) \times (\ell+1)}$ is a submatrix of $R_{[(m+1)d] \times (\ell+1)}$ and so this matrix $R_{[(m+1)d] \times (\ell+1)}$ must also have linearly independent $(\ell+1)$ columns.

As such, one can always choose an s_{m+1} such that the determinant of $R_{[(m+1)d] \times (\ell+1)}$ is non-zero thus assuring that $R_{[(m+1)d] \times (\ell+1)}$ has linearly independent columns when its rows are evaluated at complex frequencies, s_1, s_2, \dots, s_{m+1} . This completes the proof.

The above proof brings out an interesting algorithm for choosing test frequencies which no longer have to be picked in advance. For the scalar case, it can be shown that the number of test frequencies is exactly $k - \delta(r)$ (equal to the rank of the Jacobian matrix). In the general case, however, the number of required test frequencies is

less than or equal to $k - \delta(\underline{r})$.

2.3 The Global Measure

Reaching this far, we have solved the problem regarding choice of frequencies for the computation of the local measure $\delta(\underline{r})$. There is, however, one more problem. To compute $\delta(\underline{r})$ by equation (2.3) and the use of theorem 1 of section 2.2, \underline{r} should be known. This means that the value of the failed parameter vector is known or the fault is already known and isolated. If this is so, there is no need to compute $\delta(\underline{r})$ since our purpose is to do fault analysis by $\delta(\underline{r})$.

This necessitates the formulation of a global measure of testability. Such a measure is possible with the assumption that all components involved have transfer functions which are strictly rational in r .

With this assumption the difficulty can be overcome by the use of the following theorem:

Theorem 2: $\delta(r)$ is constant "almost everywhere".

Here, the term "almost everywhere" is used in the algebraic geometric sense, that is, $\delta(r)$ is constant for all r except possibly for the values lying in an algebraic variety of the parameter space R^k .

[Appendix]

Proof:

Here again we have:

$$R(s, r) = \begin{bmatrix} R_{11}(s, r) & R_{12}(s, r) & \dots & R_{1k}(s, r) \\ R_{21}(s, r) & R_{22}(s, r) & \dots & R_{2k}(s, r) \\ \vdots & \vdots & \ddots & \vdots \\ R_{d1}(s, r) & R_{d2}(s, r) & \dots & R_{dk}(s, r) \end{bmatrix} \quad (2.19)$$

which is same as equation (2.6) with the potentially variable circuit parameter r included, and k is the dimension of the parameter space R^k . Our $\delta(r)$ is the column nullity of this matrix $R(s, r)$ at r over the field of complex numbers.

Now we have to prove that the number of linearly independent column vectors $[R_{ij}(s, r); j = 1, 2, \dots, d]$ for $i = 1, 2, \dots, k$ over the field of complex numbers is constant for almost all r .

Let

$$R_{ij}(s, r) = \frac{N_{ij}(s, r)}{d(s, r)} \quad (2.20)$$

where

$$N_{ij}(s, r) = \sum_{\ell=1}^x N_{ij}^{\ell}(r) s^{\ell} \quad (2.21)$$

and $d(s, r)$ is the least common denominator of $R(s, r)$.

Now, the number of linearly independent column vectors:

$$\begin{bmatrix} R_{1j}(s,r) \\ R_{2j}(s,r) \\ \cdot \\ \cdot \\ \cdot \\ R_{dj}(s,r) \end{bmatrix} ; j = 1, 2, \dots, k \quad (2.22)$$

over the field of complex numbers is equal to the number of linearly independent column vectors:

$$\begin{bmatrix} N_{1j}(s,r) \\ N_{2j}(s,r) \\ \cdot \\ \cdot \\ \cdot \\ N_{dj}(s,r) \end{bmatrix} ; j = 1, 2, \dots, k \quad (2.23)$$

over the field of complex numbers.

However, by working directly with the coefficients $N_{ij}^{\ell}(r)$ from equation (2.21), this is also equal to the number of linearly independent column vectors of the complex matrix:

$$\begin{bmatrix}
 N_{1j}^0(r) \\
 N_{1j}^1(r) \\
 \cdot \\
 \cdot \\
 \cdot \\
 N_{1j}^x(r) \\
 \hline
 N_{2j}^0(r) \\
 N_{2j}^1(r) \\
 \cdot \\
 \cdot \\
 \cdot \\
 N_{2j}^x(r) \\
 \hline
 \cdot \\
 \cdot \\
 \cdot \\
 \hline
 N_{dj}^0(r) \\
 N_{dj}^1(r) \\
 \cdot \\
 \cdot \\
 \cdot \\
 N_{dj}^x(r)
 \end{bmatrix}
 \quad ; j = 1, 2, \dots, k \quad (2.24)$$

over the field of complex numbers.

In other words, this is equal to the rank of the $(d \times x)$ by k

complex matrix $N(r)$ given by:

$$N(r) = \begin{bmatrix} N_{11}^0(r) & N_{12}^0(r) & \dots & N_{1k}^0(r) \\ N_{11}^1(r) & N_{12}^1(r) & \dots & N_{1k}^1(r) \\ \cdot & \cdot & \dots & \cdot \\ N_{11}^x(r) & N_{12}^x(r) & \dots & N_{1k}^x(r) \\ \hline N_{21}^0(r) & N_{22}^0(r) & \dots & N_{2k}^0(r) \\ N_{21}^1(r) & N_{22}^1(r) & \dots & N_{2k}^1(r) \\ \cdot & \cdot & \dots & \cdot \\ N_{21}^x(r) & N_{22}^x(r) & \dots & N_{2k}^x(r) \\ \hline \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ \hline N_{d1}^0(r) & N_{d2}^0(r) & \dots & N_{dk}^0(r) \\ N_{d1}^1(r) & N_{d2}^1(r) & \dots & N_{dk}^1(r) \\ \cdot & \cdot & \dots & \cdot \\ N_{d1}^x(r) & N_{d2}^x(r) & \dots & N_{dk}^x(r) \end{bmatrix} \quad (2.25)$$

i.e., the dimension of the largest square non-singular submatrix of $N(r)$.

As such, the proof of this theorem reduces to showing that the rank of $N(r)$ is constant almost everywhere.

To show this, let h be the maximal rank assumed by $N(r)$ as a function of r , say at $r = \underline{r}$. Then there exists an h by h square sub-

matrix $N_h(r)$ such that its determinant, $\Delta_h(r)$ is rational in r and not identically zero. Since $\Delta_h(r)$ is rational (by assumption on the types of components), it is non-zero almost everywhere implying that $N_h(r)$ is of rank h almost everywhere. As such, the maximal value achieved for almost all values of r , showing that the rank of $N(r)$ is constant everywhere. Indeed, this constant value is equal to the maximal possible value for the rank. This in turn means that $\delta(r)$ is constant almost everywhere and thus completes the proof of the theorem.

This theorem is the key which makes our approach to formulating a measure of testability viable since it allows us to transform the local measure of testability, $\delta(r)$, into a global measure of testability δ , which is independent of the failure. In particular, we define δ as the generic value of $\delta(r)$, ie the value taken on by $\delta(r)$ almost everywhere. So defined, δ is thus independent of the choice of test frequencies employed (Theorem 1, Section 2.2) and the particular failure which one is attempting to diagnose. (Theorem 2, Section 2.3). It is thus a function only of the system under test and hence is a measure of testability for the circuit which is independent of both the failure and the test system.

2.4 An Analytic Example

Let us consider the amplifier circuit shown in figure 2.1.

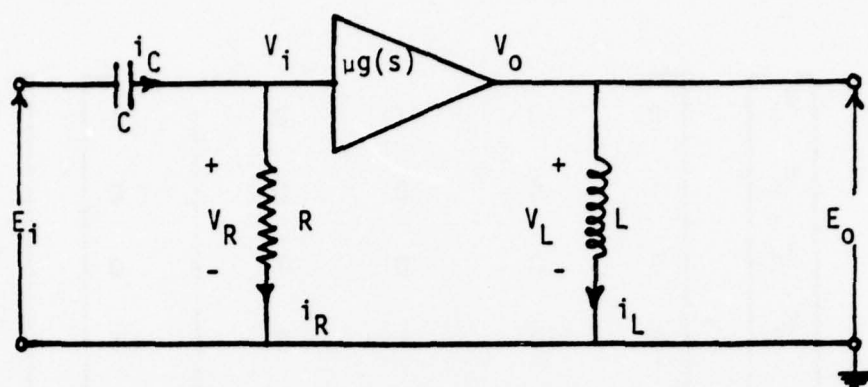


Figure 2.1 RC-coupled amplifier with an inductive load.

To illustrate the point, we shall assume in the beginning that we have access to E_o , $i_o = i_L$, $i_R = i_C$ and V_i . We can keep all four of them as available outputs in our connection model. Hence the component equation is given by:

$$\begin{bmatrix} V_o \\ i_L \\ V_C \\ i_R \end{bmatrix} = \begin{bmatrix} \mu(r)g(s) & 0 & 0 & 0 \\ 0 & 1/L(r)s & 0 & 0 \\ 0 & 0 & 1/C(r)s & 0 \\ 0 & 0 & 0 & 1/R(r) \end{bmatrix} \begin{bmatrix} V_i \\ V_L \\ i_C \\ V_R \end{bmatrix} \quad (2.26)$$

where r denotes the parameter subject to variation which in our example will be μ , R , L and C themselves, and the connection equation is given by:

$$\begin{bmatrix} V_i \\ V_L \\ i_C \\ V_R \\ -- \\ E_0 \\ i_o \\ i_R \\ V_i \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \\ -- & -- & -- & -- & -- \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} V_o \\ i_L \\ V_C \\ i_R \\ -- \\ E_i \end{bmatrix} \quad (2.27)$$

which gives us the four L matrices.

Now since δ is constant almost everywhere by Theorem 2 in Section 2.3 and is independent of the individual component values, for computational purposes, one can work with the nominal values of μ , R , L and C each equal to unity.

Using equations (2.26) and (2.27) the composite system transfer function is found with the help of equation (1.5):

$$S(s,r) = \begin{bmatrix} \frac{s(q(s)+1) + 1}{s+1} \\ \frac{q(s)}{s+1} \\ \frac{s}{s+1} \\ \frac{s}{s+1} \end{bmatrix} \quad (2.28)$$

For computation of the Jacobian matrix one can break the equation (2.1) into:

$$\frac{\partial f(Z(s, \underline{r}))}{\partial r_i} = L_{21} \frac{\partial Q(Z(s, \underline{r}))}{\partial r_i} L_{12} \quad (2.29)$$

where

$$\begin{aligned} \frac{\partial Q(Z(s, \underline{r}))}{\partial r_i} &= (I - Z(s, \underline{r})L_{11})^{-1} \frac{\partial Z(s, \underline{r})}{\partial r_i} \\ &\quad [L_{11}(I - Z(s, \underline{r})L_{11})^{-1}Z(s, \underline{r}) + I] \end{aligned} \quad (2.30)$$

The equation (2.31) is computed for each of the internal circuit parameter which is subject to variation (i.e. μ , L , C and R) and is evaluated at the nominal value.

$$\frac{\partial Q(Z(s, \underline{r}))}{\partial \mu} = \begin{bmatrix} g(s) & 0 & \frac{-g(s)}{s+1} & \frac{-g(s)}{s+1} \\ \frac{g(s)}{s} & 0 & \frac{-g(s)}{s(s+1)} & \frac{-g(s)}{s(s+1)} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.31)$$

$$\frac{\partial Q(Z(s, \underline{r}))}{\partial L} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{-g(s)}{s} & -1/s & \frac{g(s)}{s(s+1)} & \frac{g(s)}{s(s+1)} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.32)$$

$$\frac{\partial Q(Z(s,r))}{\partial C} = \begin{bmatrix} 0 & 0 & \frac{sg(s)}{(s+1)^2} & \frac{sg(s)}{(s+1)^2} \\ 0 & 0 & \frac{g(s)}{(s+1)^2} & \frac{g(s)}{(s+1)^2} \\ 0 & 0 & \frac{-s}{(s+1)^2} & \frac{-s}{(s+1)^2} \\ 0 & 0 & \frac{s}{(s+1)^2} & \frac{s}{(s+1)^2} \end{bmatrix} \quad (2.33)$$

$$\frac{\partial Q(Z(s,r))}{\partial R} = \begin{bmatrix} 0 & 0 & \frac{-g(s)}{(s+1)^2} & \frac{sg(s)}{(s+1)^2} \\ 0 & 0 & \frac{-g(s)}{(s+1)^2} & \frac{g(s)}{(s+1)^2} \\ 0 & 0 & 1/(s+1)^2 & -s/(s+1)^2 \\ 0 & 0 & s/(s+1)^2 & -s^2/(s+1)^2 \end{bmatrix} \quad (2.34)$$

Equations (2.32) through (2.35) are further used to compute

$$\frac{\partial f(Z(s,r))}{\partial \mu}, \frac{\partial f(Z(s,r))}{\partial L}, \frac{\partial f(Z(s,r))}{\partial C} \text{ and } \frac{\partial f(Z(s,r))}{\partial R} \text{ using equations}$$

(2.29) and the Jacobian matrix is computed by the use of equation (2.35).

$$J_f(r) = \begin{bmatrix} \frac{\partial f(Z(s,r))}{\partial \mu} & \frac{\partial f(Z(s,r))}{\partial L} \\ \frac{\partial f(Z(s,r))}{\partial C} & \frac{\partial f(Z(s,r))}{\partial R} \end{bmatrix} \quad (2.35)$$

which is found to be:

$$J_F(r) = \begin{bmatrix} \frac{sq(s)}{s+1} & 0 & \frac{sq(s)}{(s+1)^2} & \frac{sq(s)}{(s+1)^2} \\ \frac{q(s)}{s+1} & \frac{-q(s)}{s+1} & \frac{q(s)}{(s+1)^2} & \frac{q(s)}{(s+1)^2} \\ 0 & 0 & \frac{s}{(s+1)^2} & \frac{-s^2}{(s+1)^2} \\ 0 & 0 & \frac{s}{(s+1)^2} & \frac{s}{(s+1)^2} \end{bmatrix} \quad (2.36)$$

According to the theory outlined in previous sections of the chapter, we compute:

$$\delta = \delta(r) = 4 - \text{Number of linearly independent columns of } J_F(r) \quad (2.37)$$

for various combinations of outputs and hence those of the outputs chosen.

A tabulation of all such combinations is given in Table 2.1. A quick glance at the table (2.1) gives a measure of testability for the given circuit. For example, in case 1 and in case 2, $\delta = 0$ means that the failure can be isolated exactly, while in cases 3 through 7 and case 10, $\delta = 1$ means that it can be isolated upto an error in one parameter and so on.

TABLE 2.1
RESULTS (EXAMPLE 2.1)

Case	Number of Outputs Chosen	Output(s)	δ
1	4	E_o, i_o, i_R, V_i	0
2	3	E_o, i_o, i_R	0
3	3	i_o, i_R, V_i	1
4	3	i_R, V_i, E_o	1
5	3	V_i, E_o, i_o	1
6	2	E_o, i_o	1
7	2	i_o, i_R	1
8	2	i_R, V_i	2
9	2	V_i, E_o	2
10	2	E_o, i_R	1
11	2	i_o, V_i	2
12	1	E_o	2
13	1	i_o	2
14	1	i_R	2
15	1	V_i	3

CHAPTER III

THE MEASURE - COMPUTATIONAL ASPECTS

Given the analytic theory developed in the previous chapter, we now turn to the problem of computing the parameter δ . This is achieved by transforming the rank computation problem for $J_f(r)$ into an equivalent problem of determining the number of non-zero eigenvalues of a positive symmetric matrix: A problem for which standard computational techniques are applicable.

3.1 Computational Approach

Since the starting point for our measure of testability is the component connection model, the first aspect to be looked into is the formulation and existence of such a representation. Conditions for existence of the connection equations for an arbitrary system have been studied in the past and computer implementation are available [7-10].

When a component connection model has been formulated, it is easy to compute the $J_f(r)$ matrix of equation (2.1) by substituting the various connection matrices and the effect on the composite component transfer function corresponding to each of the potentially variable circuit parameter ie: $\frac{\partial Z(s,r)}{\partial r_i}$. As mentioned earlier, it is per-

missible to use nominal values of the various components to simplify computation of a global measure. It is interesting to note, however, that the determination of this $J_f(r)$ matrix does not involve much computer time, if the composite system transfer function $S(s,r)$ given by equation (1.5) has been computed earlier. This is the case, because no additional matrix inversion is involved.

The dimension of the $J_f(\underline{r})$ matrix given by equation (2.1) is p by q where p is the number of available output test points and q is the number of available input test points.

It is convenient, however, to make $J_f(\underline{r})$ a column vector, by stacking the various columns, under the first column. The linear independence of the matrices for various r_j can then be checked using these column vectors whose dimension is pq . Equivalently, the number of linearly independent $J_f(\underline{r})$ matrices is equal to the column rank (over the field of complex numbers) of the matrix:

$$M(s) = \left[\text{vec } \frac{\partial f}{\partial r_1} \mid \text{vec } \frac{\partial f}{\partial r_2} \mid \dots \mid \text{vec } \frac{\partial f}{\partial r_k} \right] \quad (3.1)$$

where "vec" denotes the operation of stacking columns of the $J_f(\underline{r})$ matrix into a single column.

The problem of computing the column rank of the rational matrix $M(s)$ can be transformed into an equivalent problem of computing the rank of the complex matrix:

$$M_n = \begin{bmatrix} M(s_1) \\ M(s_2) \\ \dots \\ M(s_n) \end{bmatrix} \quad (3.2)$$

where the s_j 's denote sufficiently many distinct frequencies.

Now if $M(s) V = 0$ for a complex vector V ,
then

$$M(s_j) V = 0 \quad \forall j \quad (3.3)$$

hence,

$$M_n V = \begin{bmatrix} M(s_1) \\ M(s_2) \\ \dots \\ M(s_n) \end{bmatrix} V = 0 \quad (3.4)$$

showing that,

$$\text{rank } M_n \leq \text{col rank } M(s) \quad (3.5)$$

independently of the number and value of the test frequencies.

Conversely if $M_n V = 0$ for sufficiently large n ($\geq k$ times the maximum degree of any entry in $M(s)$) then $M(s)V \equiv 0$. ie: if a rational function is zero at sufficiently many frequencies it is identically zero by the fundamental theorem of Algebra.

As such,

$$\text{col rank } M(s) \leq \text{rank } M_n \quad (3.6)$$

Therefore, from equations (3.5) and (3.6) our computational problem is reduced to the computation of:

$$\text{rank } M_n = \text{col rank } M(s) \quad (3.7)$$

where n is taken to be sufficiently large.

To further simplify the problem, the rank of M_n equals the rank of the k by k positive semi definite matrix $P = M_n^T M_n$ which is in turn equal to the number of non-zero eigenvalues of this P matrix [13].

Unfortunately, determination of the eigenvalues of a matrix is not much simpler computationally than a rank determination. In our case, however, we are not interested in a precise determination of the eigenvalues of P but only the number of non-zero eigenvalues. As such, a tridiagonalization by Householder's method followed by the computation of a Sturm sequence [13-15] may be used to compute the rank of M_n without actually computing the eigenvalues of P .

3.2 The Procedure and Test Point Selection

Based on section 3.1 a computational procedure can be outlined in the following steps:

- 1). Formulate the component connection model for the system.
- 2). Choose a range and number of test frequencies. Note that the resultant δ is independent of the choice of test frequencies so long as sufficiently many frequencies are employed. As such, we may employ real frequencies for simplifying the computations.
- 3). Formulate the $\frac{\partial f(Z(s, \underline{r}))}{\partial r_i}$ matrix for each potentially variable parameter at each test frequency.
- 4). Form the M_n matrix and P matrix.

5). Use the Sturm sequence algorithm to compute δ .

For the same circuit or system, it is possible to carry out the above analysis with various (or all) possible combinations of test inputs and outputs. It is also possible to give cost values to each of these test inputs and outputs depending upon factors such as ease of accessibility, effect on system and test equipment required, etc. Then one can determine an optimum choice of test points compatible with a desired degree of testability.

The above procedure is used to develop the computer implementation explained in more detail in Chapter IV.

CHAPTER IV

COMPUTER IMPLEMENTATION

Now that we have developed the mathematical theory and the computational aspects behind our measure of testability let us come to its specific implementation in a FORTRAN Program. In this chapter, we present some details on the actual computer implementation of the algorithm developed in Chapter III earlier. The algorithm has been implemented in the Electronic Circuit Measurability Package (ECMP) by the author for purpose of verifying the computational feasibility of δ on a CDC 1604 computer and to run the examples to be presented in Chapter V. No attempt is, however, made in the chapter to describe the said package, rather a flow chart and the salient features on computer implementation are presented in some detail. Specific details on the ECMP package are available in reference [17].

4.1 Inputs

As mentioned earlier, the computation of the measure δ requires the input of a complete component connection model. However, when working on the computer for selection of proper test points, certain additional inputs are needed. A complete listing of the inputs desired is given below:

1) The L_{ij} Matrices: These represent the connection structure of the circuit or system for the available number of inputs and outputs under test for selection of test points. Their dimensions also indicate the number of available inputs and outputs.

2) The Component Matrix: (Z) This represents the number and types of components in the system. It may be noted that since it has

frequency(s) dependent components and since we have to vary frequencies in the algorithm, it is convenient to input it in the form of a subroutine in the variable s with the calling program specifying s .

3) The Sensitivity Matrix: $(\frac{\partial Z}{\partial r})$ This will be used in computation of the Jacobian matrix. Similar remarks regarding multifrequency computation in the form of a subroutine apply here also.

4) The Choice of Frequencies: Though this is theoretically not a critical problem, yet computationally it proves to be significant. In order to work accurately and efficiently with the finite word size of the computer, the proper choice of frequencies is important. For this purpose, a starting frequency and a step size may be needed as an input for a sequential choice of frequencies.

5) Number of Frequencies: This should be "sufficient" as detailed in Section 3.1.

6) Number of Potentially Variable Circuit Parameters: This will determine the size of the Jacobian Matrix.

7) Maximum Number of Cases to be Analyzed: This depends on the number of inputs and outputs available and is given by:

$$({}_2\text{Number of Inputs}_{-1}) \times ({}_2\text{Number of Outputs}_{-1})$$

8) Maximum number of iterations to be done at an assumed interval (say 10^{-1}) starting from an assumed initial test eigenvalue (say 1.0) for Sturm Sequence computation.

9) Cost of each test point: This will be needed for optimization purposes and is given in the form of two vectors separately for inputs and outputs.

Needless to say, the proper dimensioning of all the arrays including the working one, if any, to their maximum size is also required as part of the input. In addition, some information on the options exercised in use of formats, outputs and printouts required may be needed depending upon the specific implementation.

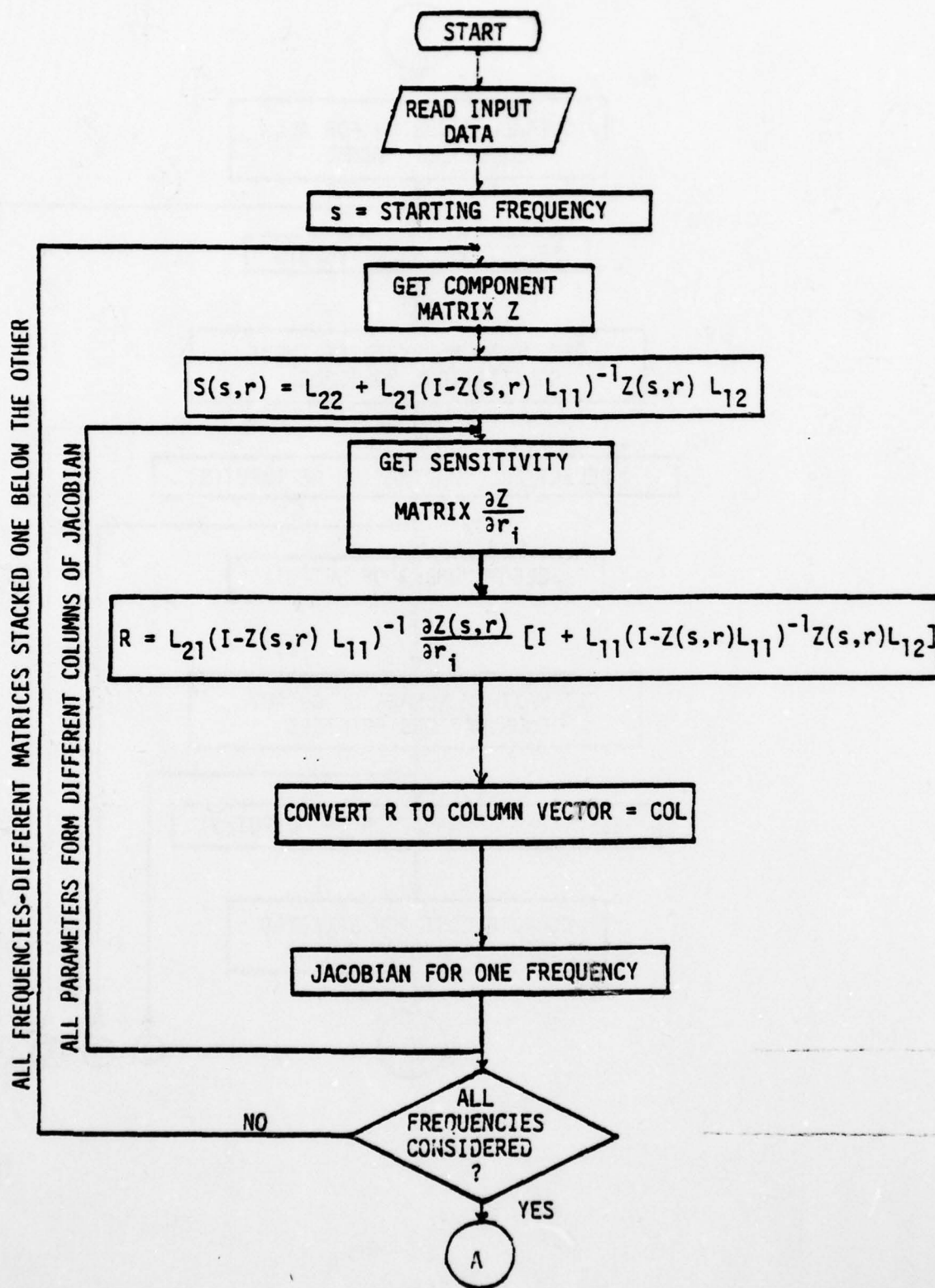
4.2 Outputs

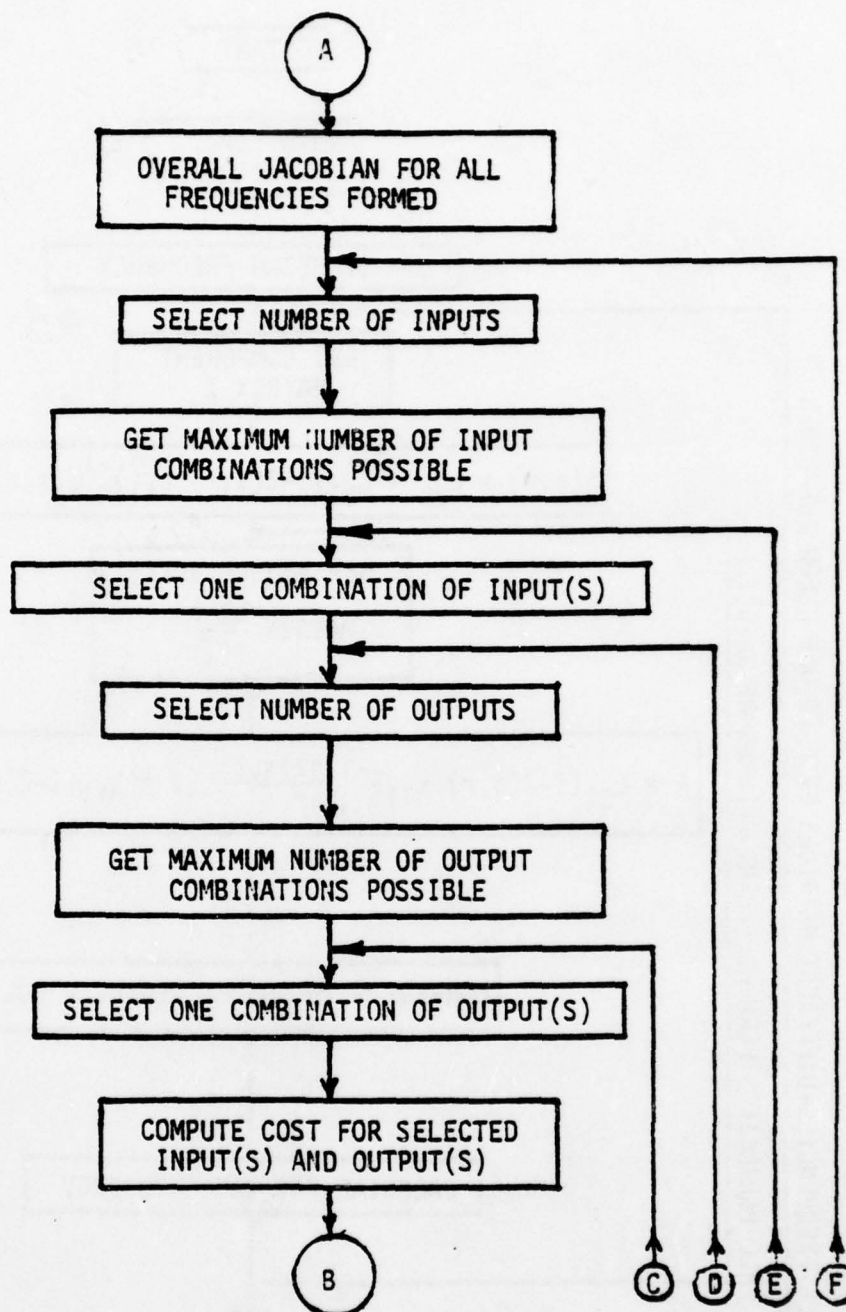
The outputs desired from the program depend upon the user requirements. For example, one may choose to get information on the composite system transfer function, the Jacobian matrix or the Sturm sequences. Alternatively, one may choose to have the minimum possible information. Invariably, we shall need a printout of the designation of the selected number of input(s) and output(s) combination, corresponding measure δ and the cost involved. This may be needed for each combination if no optimization is included in the implementation.

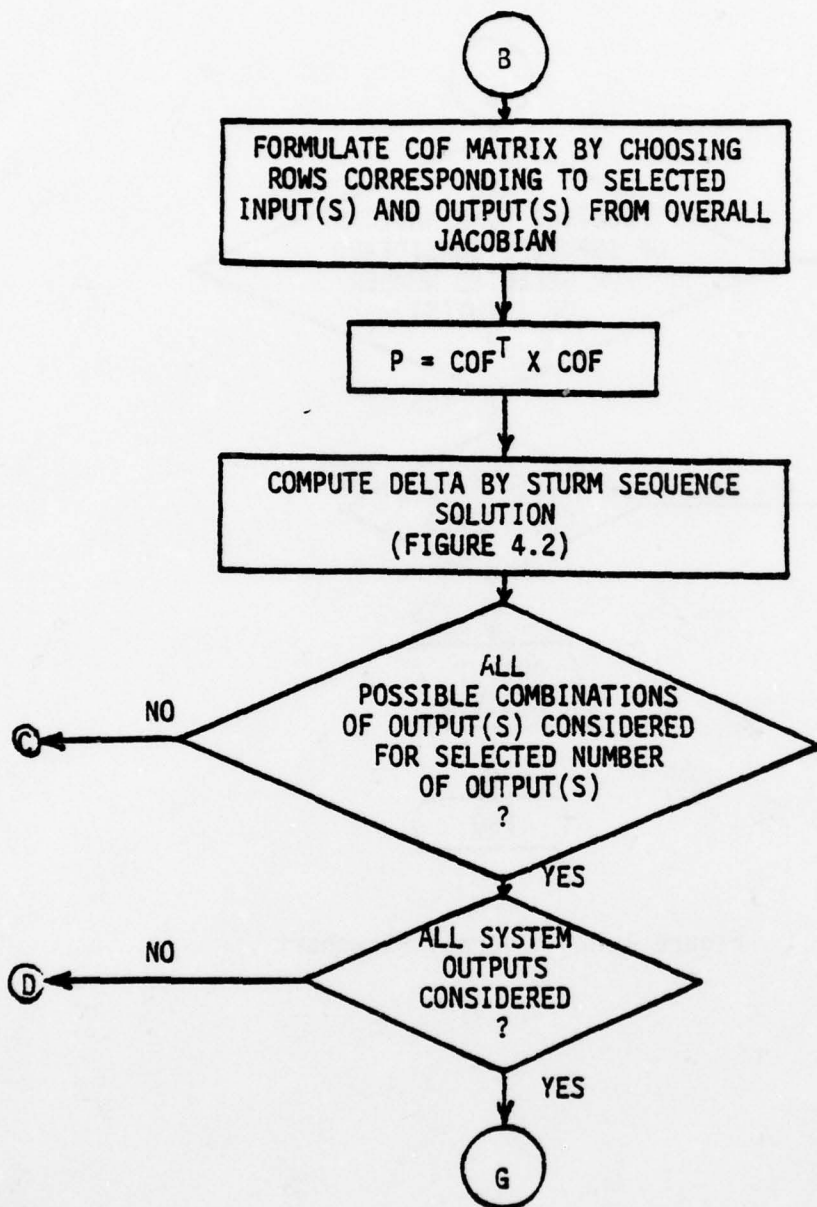
4.3 Overall Flowchart

The overall flowchart is shown in figure 4.1. This chart gives the computer implementation without restricting itself to a specific implementation in FORTRAN or using program symbols.

The whole computation is divided into two major sections. One section of the flowchart concerns computation of the Composite System Transfer Function (if desired) and the overall Jacobian formulation for the selected set of frequencies and potentially variable circuit parameters. The other portion involves computation of cost and the delta measure for each combination of inputs and outputs.







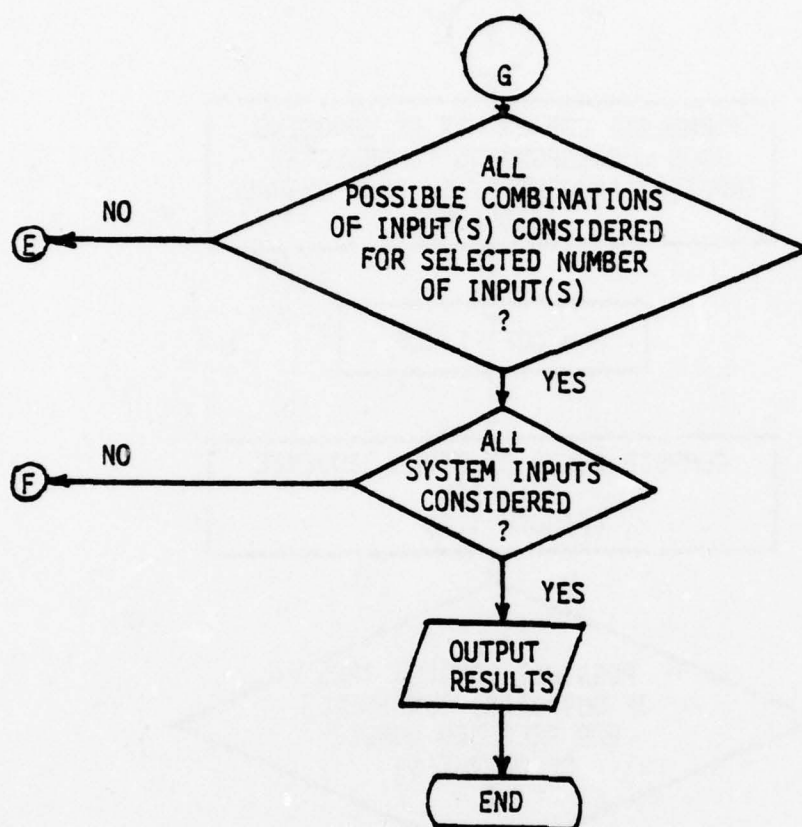


Figure 4.1. Overall Flowchart

Computation of the Composite System transfer function and the overall Jacobian Matrix (for all frequencies) is relatively straightforward. After reading the input data as explained in Section 4.1, the component matrix Z is computed corresponding to the chosen frequency. Then composite system transfer function may be computed by the use of equation (1.5) within a few matrix operations. Next, the sensitivity matrix is computed for one variable circuit parameter. This leads to the computation of corresponding entries in the Jacobian matrix by use of equation (2.1). It may be noted that no additional matrix inversion is involved at this stage, which results in saving in computer time. This will be further converted to a column vector. Similar column vectors corresponding to each potentially variable circuit parameters, stacked side by side give the Jacobian matrix for a single frequency. The overall Jacobian matrix is formed by placing all such matrices for each chosen frequency columnwise one below the other.

For computation of delta and the associated cost, the overall Jacobian formulated in the first part is used. This is done by picking a certain number of inputs (starting with one) and a certain number of outputs (also starting with one) and generating all possible combinations of this selection. Corresponding to the set of inputs and outputs selected, one can then pick the appropriate rows of the overall Jacobian matrix and formulate the COF matrix. As explained in the procedure in Section 3.3, we can now make a smaller dimension matrix P which can serve the purpose of this computation. This P matrix is tested for a Sturm Sequence Solution which gives the value of δ . At the same time, one can determine the cost of the chosen test configuration by

AD-A060 941

TEXAS TECH UNIV LUBBOCK INST FOR ELECTRONICS SCIENCE
FAULT ANALYSIS IN ELECTRONIC CIRCUITS AND SYSTEMS. II.(U)
JAN 78 R SAEKS, N SEN, H M CHEN, K S LU

F/6 9/3

N00014-75-C-0924

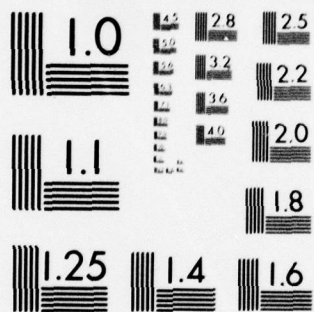
UNCLASSIFIED

NL

3 OF 4

AD
A060941





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

the values assigned at the input.

The whole procedure in the above paragraph is repeated now for a different number of outputs keeping the number of inputs the same. When all these combinations are over, the number of inputs is changed and delta and cost again computed for each combination of output(s). When all possible cases have been considered this way, one is ready to output the results.

4.3 Subroutines

Several subroutines are needed in order to do the computations repetively. Apart from the user supplied data subroutines these include the usual matrix subroutines, (addition, subtraction, multiplication, inverse, identity and zero, ect.), the subroutine for converting a matrix into a column vector, subroutine to compute factorial of a number, a subroutine to compute maximum number of combinations possible for a given set of inputs and outputs, a subroutine to formulate a new combination every time and a subroutine to compute the value of delta by Sturm Sequence Solution. As all other subroutines are relatively straightforward, we may just look into the computation of delta by the Sturm Sequence Subroutine.

Further, since this also has two standard steps involved ie:, that of tri-diagonalization of the positive definite matrix by Householder's method and the computation of Sturm Sequence [15], we may only look at the computations of delta from the Sturm Sequence thus formed.

A simplified flowchart is shown in Figure 4.2.

A Sturm Sequence $1, p_1(\lambda), \dots, p_n(\lambda)$ is computed from the tridiagonal matrix for a test eigenvalue $\lambda = 1$ (say). The total number

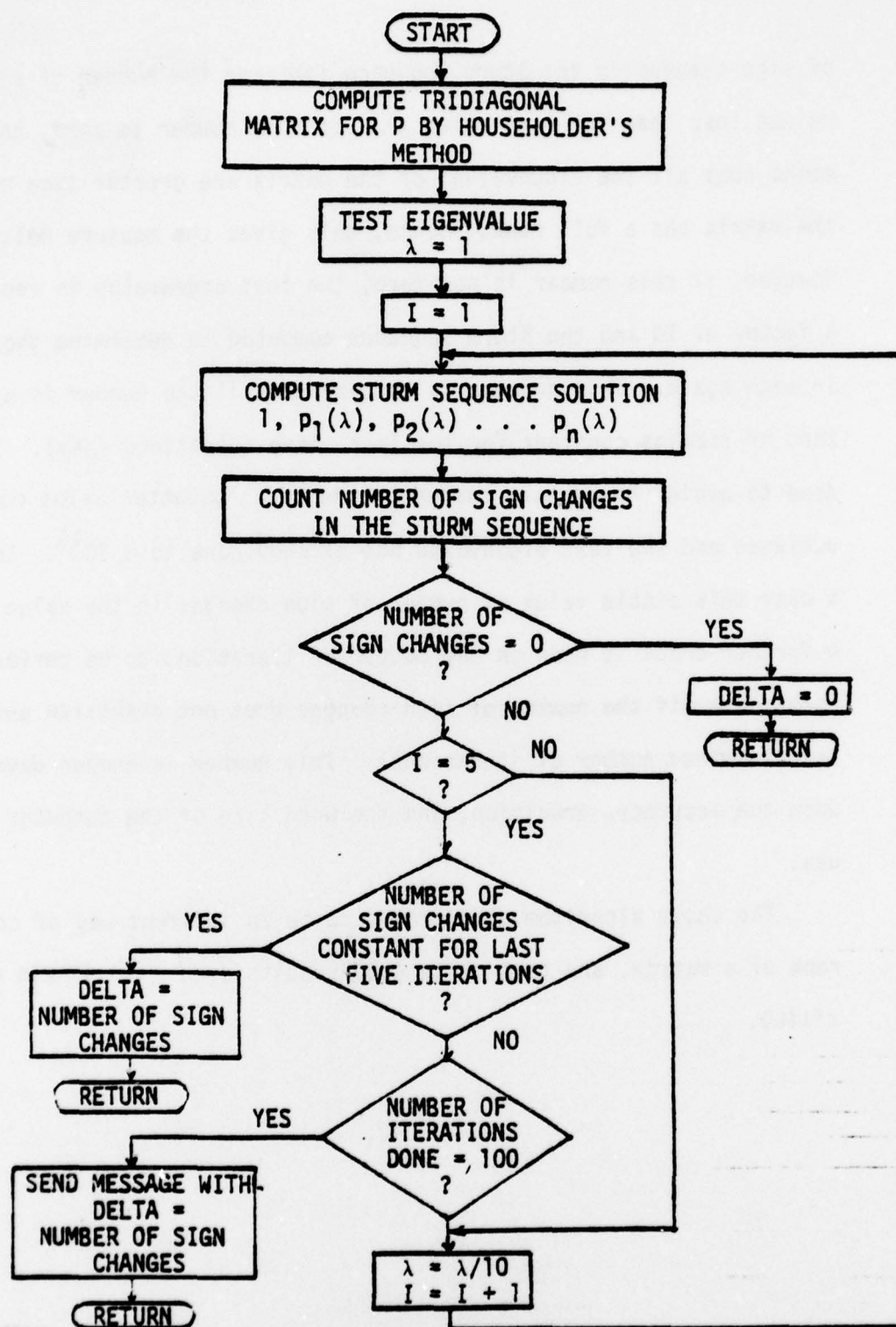


Figure 4.2. Flowchart for Computation of Delta

of sign changes in the Sturm Sequence indicate the number of eigenvalues less than the test value λ . If this number is zero, this means that all the eigenvalues of the matrix are greater than unity or the matrix has a full rank. Hence, this gives the measure $\delta = 0$. However, if this number is non-zero, the test eigenvalue is reduced by a factor of 10 and the Sturm Sequence computed to determine the changes in sign again. This process is continued until the number is either zero or remains constant for the last five iterations (say). This is done to avoid unnecessary computer time if a no better value can be achieved and the test eigenvalue has already gone to $\leq 10^{-5}$. In such a case this stable value of number of sign changes is the value of δ . A further check is made on the number of iterations to be performed in worst case (if the number of sign changes does not stabilize until a predetermined number of iterations). This number is chosen depending upon the accuracy, precision, and the word size of the computer under use.

The above algorithm also proves to be an indirect way of computing rank of a matrix, and gave quite good results even with single precision.

CHAPTER V

EXAMPLES

In this chapter we will present a number of examples. These range from transistor circuits, passive and active filters, oscillator and finally to ladder networks. All of them were run on computer package ECMP for computation of the δ parameter and the cost of each alternative.

5.1 One Stage Transistor Amplifier

As a first example, consider the single stage NPN RC-Coupled transistor amplifier shown in Figure 5.1 and its hybrid- π equivalent circuit shown in Figure 5.2. [19] Assuming four outputs available (for comparison purposes) and a single input, a set of connection equations for the equivalent circuit are given in equation 5.1 and the corresponding set of component equations are given in equation 5.2. Since it is not physically possible to isolate a fault in one of the two parallel resistors while they are still connected, a parallel combination R_a' of the two resistors R_a and R_b is considered in the equations, thus giving rise to 12 components. As δ is constant almost everywhere and thus independent of the actual component values, for computational purposes one can take the nominal value of all the 12 components to be unity. Also 15 real frequencies with a starting frequency of 1 and a step size of unity will serve the purpose. The maximum number of possible cases for various combinations of test inputs and outputs is 15.

As per the procedure outlined in section 3.2, a Sturm sequence

solution was used to compute δ in each possible case and a tabulation of the values obtained is given in Table 5.2. These include costs assigned (for illustrative purpose) to various inputs and outputs according to Table 5.1.

Table 5.2 gives the measure of testability δ and the associated cost for each case. For example, for a failure to be isolated exactly we require $\delta=0$, it will then be cheapest to select case 5 (with input 1 and outputs 1 and 2) at a cost of 4. Further, if an error in a single parameter is acceptable ie $\delta=1$, case 6 will also yield a cost of 4. However, if a $\delta=2$ (with input 1 and output 2) and 3 (with input 1 and output 3) prove to be equally attractive with a cost of 3 each, etc.

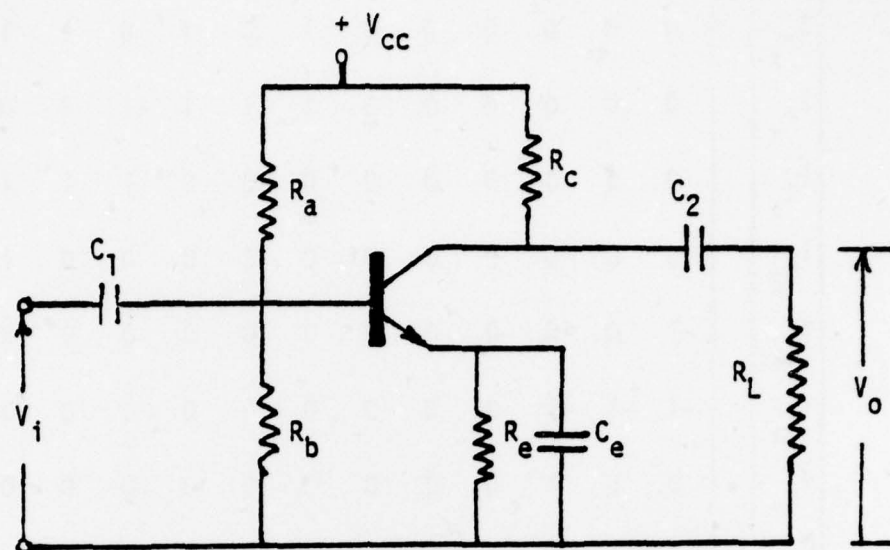


Figure 5.1 One Stage Transistor Amplifier

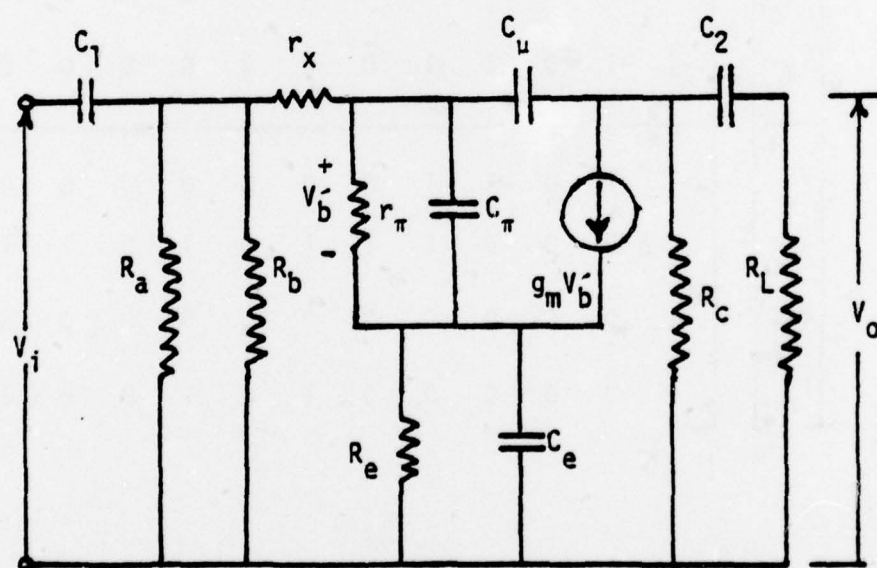


Figure 5.2 Amplifier Equivalent Circuit

$$\begin{bmatrix} I_{C_1} \\ I_{r_x} \\ I_{r_\pi} \\ I_{C_\mu} \\ I_{C_2} \\ V_{R_a}' \\ V_{R_e} \\ V_{C_\pi} \\ V_{C_e} \\ V_{g_m} \\ V_{R_c} \\ V_{R_L} \\ \hline V_o \\ I_{C_1} \\ V_{R_a}' \\ I_e \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline -1 & -1 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ \hline 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} V_{C_1} \\ V_{r_x} \\ V_{r_\pi} \\ V_{C_\mu} \\ V_{C_2} \\ I_{R_a}' \\ I_{R_e} \\ I_{C_\pi} \\ I_{C_e} \\ I_{g_m} \\ I_{R_c} \\ I_{R_L} \\ \hline V_i \end{bmatrix}$$

(5.1)

$$\begin{bmatrix} V_{C1} \\ V_{r_x} \\ V_{r_\pi} \\ V_{C\mu} \\ V_{C2} \\ I_{R'_a} \\ I_{R_e} \\ I_{C_\pi} \\ I_{C_e} \\ I_{g_m} \\ I_{R_C} \\ I_{R_L} \end{bmatrix} = \begin{bmatrix} K_{1s} & & & & & & & & & & & \\ & r_x & & & & & & & & & & \\ & & r_\pi & & & & & & & & & \\ & & & 1/C_\mu & & & & & & & & \\ & & & & K_{2s} & & & & & & & \\ & & & & & 1/R'_a & & & & & & \\ & & & & & & 1/R_e & & & & & \\ & & & & & & & C_\pi & & & & \\ & & & & & & & & C_e & & & \\ & & & & & & & & & g_m & 0 & \\ & & & & & & & & & & 1/R_C & \\ & & & & & & & & & & & 1/R_L \end{bmatrix} \begin{bmatrix} I_{C1} \\ I_{r_x} \\ I_{r_\pi} \\ I_{C\mu} \\ I_{C2} \\ V_{R'_a} \\ V_{R_e} \\ V_{C_\pi} \\ V_{C_e} \\ V_{g_m} \\ V_{R_C} \\ V_{R_L} \end{bmatrix}$$

TABLE 5.1
INPUTS AND OUTPUTS (Example 1)

<u>Input</u>			<u>Outputs</u>		
Number Designation	Type Designation	Cost	Number Designation	Type Designation	Cost
1	V_f	1	1	V_o	1
			2	I_{C1}	2
			3	V_{R_a}	2
			4	I_e	3

TABLE 5.2
RESULTS (Example 5.1)

Case	No. of Inputs Chosen	No. of Outputs Chosen	No. Designation of Input Chosen	No. Designation of Output Chosen	Cost	Delta
1	1	1	1	1	2	3
2	1	1	1	2	3	2
3	1	1	1	3	3	2
4	1	1	1	4	4	3
5	1	2	1	1,2	4	0
6	1	2	1	1,3	4	1
7	1	2	1	1,4	5	0
8	1	2	1	2,3	5	2
9	1	2	1	2,4	6	1
10	1	2	1	3,4	6	0
11	1	3	1	1,2,3	6	0
12	1	3	1	1,2,4	7	0
13	1	3	1	1,3,4	7	0

14	1	3	1	2,3,4	8	0
15	1	4	1	1,2,3,4	9	0

5.2 Transistor with Load

The second example illustrates the application to a transistor with load shown in Figure 5.3. Figure 5.4 shows the hybrid- π equivalent circuit. Assuming no variations in the load (it is not internal to the transistor circuit), and viewing R_L as a connection one set of component equations is given by equations (5.3) and (5.4).

The other data used is:

Nominal value of each component	= 1
Number of components	= 5
Number of inputs	= 2
Number of outputs	= 2
Number of potentially variable parameters	= 5
Number of frequencies used	= 15
Starting frequency	= 1
Step size for frequency	= 1

Table 5.3 shows the cost values assigned to the inputs and outputs and Table 5.4 gives the results for all the 9 possible combinations of the available inputs and outputs. Here it can be seen from Table 5.4 that no combination can provide a $\delta=0$. For $\delta=1$, it is cheapest to select case 2 (Input 1 and Output 2).

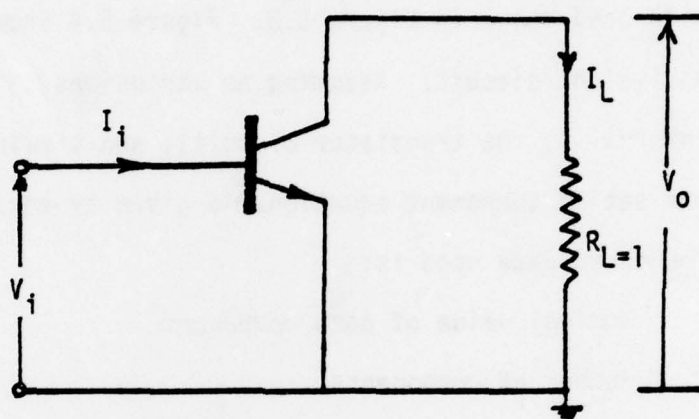


Figure 5.3 Transistor with Load

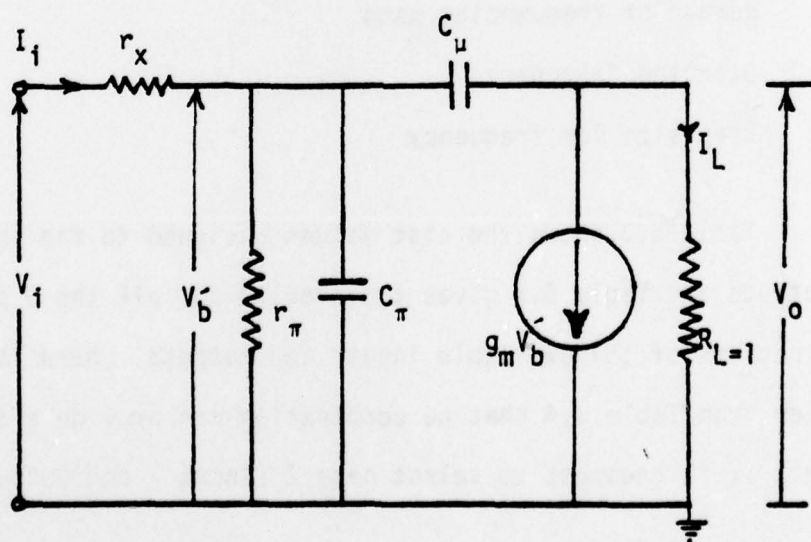


Figure 5.4 Transistor Equivalent Circuit

$$\begin{bmatrix} V_{r_x} \\ V_{c_u} \\ V_{c_\pi} \\ I_{r_\pi} \\ I_{g_m} \\ \hline V_o \\ I_f \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} I_{r_x} \\ I_{c_u} \\ I_{c_\pi} \\ V_{r_\pi} \\ V_{g_m} \\ \hline V_f \\ I_L \end{bmatrix}$$

(5.3)

$$\begin{bmatrix} I_{r_x} \\ I_{c_u} \\ I_{c_\pi} \\ V_{r_\pi} \\ V_{g_m} \end{bmatrix} = \begin{bmatrix} \cancel{I_{r_x}} & & & & 0 \\ & C_u s & & & \\ & & C_\pi s & & \\ & & & r_\pi & \\ & 0 & & & \cancel{I_{g_m}} \end{bmatrix} \begin{bmatrix} V_{r_x} \\ V_{c_u} \\ V_{c_\pi} \\ I_{r_\pi} \\ I_{g_m} \end{bmatrix}$$

(5.4)

TABLE 5.3
INPUTS AND OUTPUTS (Example 5.2)

<u>Inputs</u>			<u>Outputs</u>		
Number Designation	Type Designation	Cost	Number Designation	Type Designation	Cost
1	V_i	1	1	V_o	2
2	I_L	3	2	I_i	3

TABLE 5.4
RESULTS (Example 5.2)

Case	No. of Inputs Chosen	No. of Outputs Chosen	No. Designation of Inputs Chosen	No. Designation of Outputs Chosen	Cost	Delta
1	1	1	1	1	3	5
2	1	1	1	2	4	1
3	1	2	1	1,2	6	1
4	1	1	2	1	5	5
5	1	1	2	2	6	1
6	1	2	2	1,2	8	1
7	2	1	1,2	1	6	5
8	2	1	1,2	2	7	1
9	2	2	1,2	1,2	9	1

5.3 Band Elimination Passive Filter

The filter is shown in Figure 5.5 with a set of component and connection equations given by equations (5.5) and (5.6). The other data used is:

Nominal value of each component	= 1
Number of components	= 8
Number of inputs	= 1
Number of outputs	= 4
Number of potentially variable parameters	= 8
Number of frequencies used	= 15
Starting frequency	= 1
Step size for frequency	= 1

For the costs assigned to inputs and outputs as per Table 5.5 the results for all 15 cases are shown in Table 5.6.

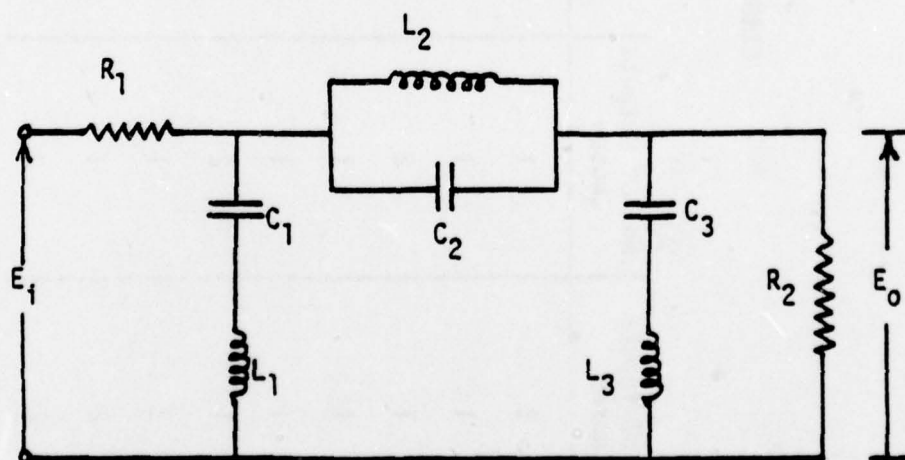


Figure 5.5 Band Elimination Passive Filter

$$\begin{bmatrix} V_{R_1} \\ I_{C_1} \\ I_{L_1} \\ V_{L_2} \\ V_{C_2} \\ I_{C_3} \\ I_{L_3} \\ V_{R_2} \\ \hline E_0 \\ I_{C_1} \\ V_{C_2} \\ I_{C_3} \end{bmatrix} = \begin{bmatrix} 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} I_{R_1} \\ V_{C_1} \\ V_{L_1} \\ I_{L_2} \\ I_{C_2} \\ V_{C_3} \\ V_{L_3} \\ I_{R_2} \\ \hline E_1 \end{bmatrix}$$

(5.5)

$$\begin{bmatrix} i_{R1} \\ v_{C1} \\ v_{L1} \\ i_{L2} \\ i_{C2} \\ v_{C3} \\ v_{L3} \\ i_{R2} \end{bmatrix} = \begin{bmatrix} \chi_1 & & & & & & & 0 \\ & \chi_{1s} & & & & & & \\ & & \chi_{1s} & & & & & \\ & & & \chi_{2s} & & & & \\ & & & & c_2s & & & \\ & & & & & \chi_{3s} & & \\ & & & & & & L_3s & \\ & & & & & & & \chi_{R2} \end{bmatrix} \begin{bmatrix} v_{R1} \\ i_{C1} \\ i_{L1} \\ v_{L2} \\ v_{C2} \\ i_{C3} \\ i_{L3} \\ v_{R2} \end{bmatrix} \quad (5.6)$$

TABLE 5.5
INPUTS AND OUTPUTS (Example 5.3)

<u>Input</u>			<u>Outputs</u>		
Number Designation	Type Designation	Cost	Number Designation	Type Designation	Cost
1	E_1	1	1	E_0	2
			2	I_{C_1}	4
			3	V_{C_2}	3
			4	I_{C_3}	4

TABLE 5.6
RESULTS (Example 5.5)

Case	No. of Inputs Chosen	No. of Outputs Chosen	No. Designation of Inputs Chosen	No. Designation of Outputs Chosen	Cost	Delta
1	1	1	1	1	3	1
2	1	1	1	2	5	2
3	1	1	1	3	4	1
4	1	1	1	4	5	1
5	1	2	1	1,2	7	0
6	1	2	1	1,3	6	0
7	1	2	1	1,4	7	0
8	1	2	1	2,3	8	0
9	1	2	1	2,4	9	0
10	1	2	1	3,4	8	1
11	1	3	1	1,2,3	10	0
12	1	3	1	1,2,4	11	0
13	1	3	1	1,3,4	10	0
14	1	3	1	2,3,4	12	0
15	1	4	1	1,2,3,4	14	0

5.4 Low Pass Active Filter

The filter shown in Figure 5.6 with a set of component and connection equations given by equations (5.7) and (5.8). The other data used is:

Nominal value of each component	= 1
Transfer function of the operational amplifier $\mu K(s)$	= $\frac{1}{(s+1)(s+2)}$
Number of components	= 6
Number of inputs	= 1
Number of outputs	= 5
Number of potentially variable parameters	= 8 (including μ and the two poles for the operational amplifier)
Number of frequencies used	= 15

For the costs assigned to the input and outputs as per Table 5.7 the results for all 31 cases are shown in Table 5.8. It may be noted that the same example run with different starting frequencies (SFREQ) and step size (STEP) gave different results. This is due to the limited accuracy in the computer and illustrates that a judicious choice of frequency, however, is important (depending upon the components involved and the nature and behavior of the circuit) to obtain best results within the same numerical accuracy limits. Here the run, with a

starting frequency of 0.5 and a step size of 0.2 gave the best results for the measure.

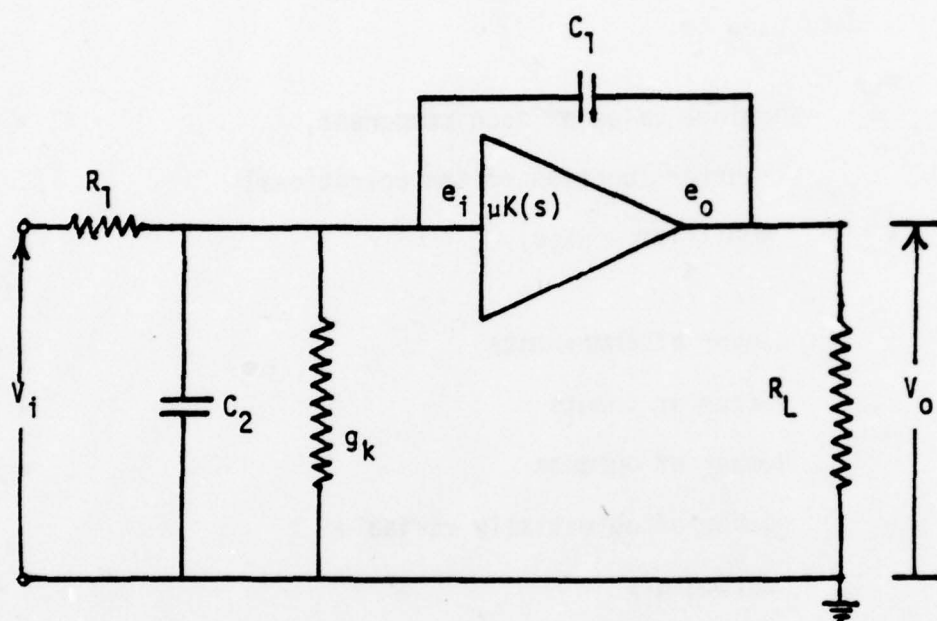


Figure 5.6 Low-Pass Active Filter

$$\begin{bmatrix} v_{R_1} \\ v_{C_1} \\ i_{C_2} \\ v_{g_k} \\ e_i \\ v_{R_L} \\ \hline v_o \\ v_{R_1} \\ v_{C_1} \\ v_{C_2} \\ e_i \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 0 & | & 1 \\ 0 & 0 & 1 & 0 & -1 & 0 & | & 0 \\ -1 & -1 & 0 & 0 & 0 & 0 & | & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & | & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & | & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & | & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 & | & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & | & 1 \\ 0 & 0 & 1 & 0 & -1 & 0 & | & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & | & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & | & 0 \end{bmatrix} \begin{bmatrix} i_{R_1} \\ i_{C_1} \\ v_{C_2} \\ i_{g_k} \\ e_o \\ i_{R_L} \\ \hline v_i \end{bmatrix}$$

(5.7)

$$\begin{bmatrix} I_{R_1} \\ I_{C_1} \\ V_{C_2} \\ I_{g_k} \\ e_o \\ I_{R_L} \end{bmatrix} = \begin{bmatrix} \cancel{1/R_1} & & & & & \\ & C_1 s & & & & \\ & & \cancel{1/C_2 s} & & & \\ & & & g_k & & \\ & & & & \mu K(s) & \\ & 0 & & & & \cancel{1/R_L} \end{bmatrix} \begin{bmatrix} V_{R_1} \\ V_{C_1} \\ I_{C_2} \\ V_{g_k} \\ e_i \\ V_{R_L} \end{bmatrix} \quad (5.8)$$

TABLE 5.7
INPUTS AND OUTPUTS (Example 5.4)

<u>Input</u>			<u>Outputs</u>		
Number Designation	Type Designation	Cost	Number Designation	Type Designation	Cost
1	V_i	1	1	V_o	2
			2	V_{R_1}	3
			3	V_{C_1}	2
			4	I_{C_2}	5
			5	e_i	2

TABLE 5.8
RESULTS (Example 5.4)

Case	No. of Inputs Chosen	No. of Outputs Chosen	No. Designation of Inputs Chosen	No. Designation of Outputs Chosen	Cost	Delta		
						SFREQ=1 STEP=1	SFREQ=0.5 STEP=0.1	SFREQ=0.5 STEP=0.2
1	1	1	1	1	3	2	2	3
2	1	1	1	2	4	2	3	3
3	1	1	1	3	3	3	2	3
4	1	1	1	4	6	2	3	3
5	1	1	1	5	3	2	3	3
6	1	2	1	1,2	6	3	2	2
7	1	2	1	1,3	5	3	3	3
8	1	2	1	1,4	8	2	1	1
9	1	2	1	1,5	5	3	2	2
10	1	2	1	2,3	6	3	3	1
11	1	2	1	2,4	9	1	3	2
12	1	3	1	2,5	6	4	2	3
13	1	2	1	3,4	8	2	1	2
14	1	2	1	3,5	5	3	3	2

15	1	2	1	4,5	8	1	3	2
16	1	3	1	1,2,3	8	1	3	1
17	1	3	1	1,2,4	11	1	2	2
18	1	3	1	1,2,5	8	2	2	2
19	1	3	1	1,3,4	10	1	1	2
20	1	3	1	1,3,5	7	1	3	2
21	1	3	1	1,4,5	10	1	2	2
22	1	3	1	2,3,4	11	1	1	2
23	1	3	1	2,3,5	8	2	1	1
24	1	3	1	2,4,5	11	1	2	2
25	1	3	1	3,4,5	10	2	1	2
26	1	4	1	1,2,3,4	13	1	2	1
27	1	4	1	1,2,3,5	10	1	3	3
28	1	4	1	1,2,4,5	13	2	1	2
29	1	4	1	1,3,4,5	12	1	1	1
30	1	4	1	2,3,4,5	13	2	1	2

31 | 1 | 5 | 1 | 1,2,3,4,5 | 15 | 1 | 1 | 2

5.5 Wien-Bridge Oscillator

The oscillator is shown in Figure 5.7 and its equivalent circuit is shown in Figure 5.8. Note that the equivalent circuit also includes the test generator input V_i and the load resistance R_L . Since generator resistance is external to the circuit no variation in r_g is assumed. A set of component connection equations is given in equations (5.9) and (5.10). The other data used is:

Nominal value of each component except

R_a and R_b	= 1
Nominal value of R_a and R_b	= 2
Number of components	= 7
Number of inputs	= 1
Number of outputs	= 4
Number of potentially variable parameters	= 7 (including R_a and R_b for the operational amplifier but excluding r_g)
Number of frequencies used	= 15

For the costs assigned to inputs and outputs as per Table 5.9, the results for all cases are shown in Table 5.10. Here again, the results are shown for three different sets of frequencies and the best results are obtained with a starting frequency of 0.1 and a step size of 0.2.

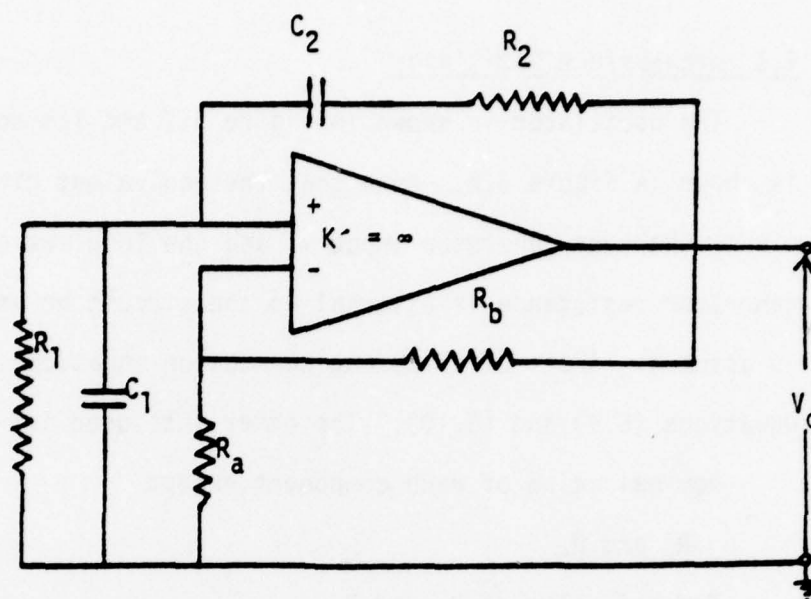


Figure 5.7 Wien Bridge Oscillator

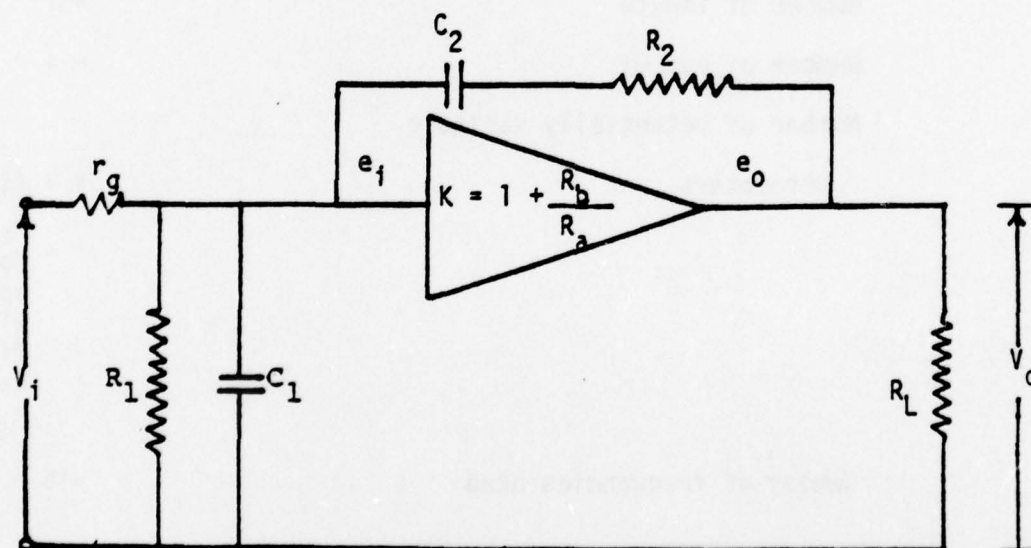


Figure 5.8 Oscillator Equivalent Circuit

$$\begin{bmatrix} I_{r_g} \\ V_{R_1} \\ V_{C_1} \\ I_{C_2} \\ V_{R_2} \\ e_i \\ V_{R_L} \\ \hline V_o \\ I_{R_g} \\ I_{C_2} \\ V_{R_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ \hline 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} V_{r_g} \\ I_{R_1} \\ I_{C_1} \\ V_{C_2} \\ I_{R_2} \\ e_o \\ I_{R_L} \\ \hline V_i \end{bmatrix}$$

(5.9)

$$\begin{bmatrix} v_{r_g} \\ I_{R_1} \\ I_{C_1} \\ v_{C_2} \\ I_{R_2} \\ e_o \\ I_{R_L} \end{bmatrix} = \begin{bmatrix} r_g & & & & & & \\ & \cancel{1/R_1} & & & & & \\ & & C_1 s & & & & \\ & & & \cancel{1/C_2 s} & & & \\ & & & & \cancel{1/R_2} & & \\ & & & & & K & \\ & & & & & & \cancel{1/R_L} \\ & 0 & & & & & \end{bmatrix} \begin{bmatrix} I_{r_g} \\ v_{R_1} \\ v_{C_1} \\ I_{C_2} \\ v_{R_2} \\ e_i \\ v_{R_L} \end{bmatrix}$$

(5.10)

TABLE 5.9
INPUTS AND OUTPUTS (Example 5.5)

<u>Input</u>			<u>Outputs</u>		
Number Designation	Type Designation	Cost	Number Designation	Type Designation	Cost
1	V_i	2	1	V_o	2
			2	I_{r_g}	4
			3	I_{C_2}	5
			4	V_{R_2}	3

TABLE 5.10
RESULTS (Example 5.5)

Case	No. of Inputs Chosen	No. of Outputs Chosen	No. Designation of Inputs Chosen	No. Designation of Outputs Chosen	Cost	Delta		
						SFREQ=1 STEP=1	SFREQ=0.5 STEP=0.1	SFREQ=0.1 STEP=0.2
1	1	1	1	1	4	2	2	2
2	1	1	1	2	6	3	3	3
3	1	1	1	3	7	3	4	3
4	1	1	1	4	5	3	3	4
5	1	2	1	1,2	8	2	2	2
6	1	2	1	1,3	9	3	2	1
7	1	2	1	1,4	7	2	2	2
8	1	2	1	2,3	11	3	3	3
9	1	2	1	2,4	9	3	3	3
10	1	2	1	3,4	10	3	3	3
11	1	3	1	1,2,3	13	2	2	1
12	1	3	1	1,2,4	11	2	2	1
13	1	3	1	1,3,4	12	2	2	2

14	1	3	1	2,3,4	14	3	3	3
15	1	4	1	1,2,3,4	16	2	2	2

5.6 Four Component RC Ladder Network

The circuit is shown in Figure 5.9 and a set of connection equation is given by equations (5.11) and (5.12). The other data used is:

Nominal value of each component	= 1
Number of components	= 4
Number of inputs	= 1
Number of outputs	= 2
Number of potentially variable parameters	= 4
Number of frequencies used	= 15
Starting frequency	= 1
Step size	= 1

For the costs assigned to inputs and outputs as per Table 5.11, the results for the three cases are shown in Table 5.12.

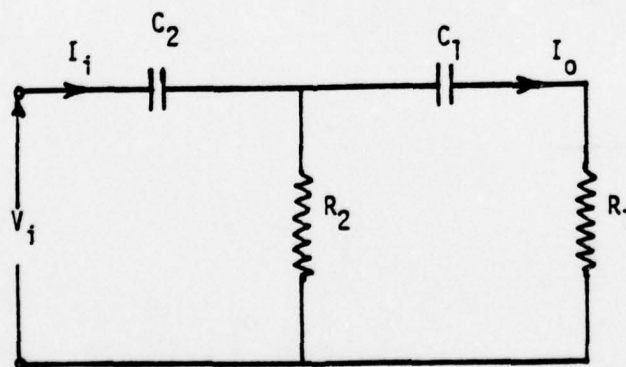


Figure 5.9 Four Component RC Ladder Network

$$\begin{bmatrix} V_{R_1} \\ V_{R_2} \\ I_{C_1} \\ I_{C_2} \\ \hline I_i \\ I_o \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 & -1 & | & 1 \\ 0 & 0 & 0 & -1 & | & 1 \\ 1 & 0 & 0 & 0 & | & 0 \\ 1 & 1 & 0 & 0 & | & 0 \\ \hline 1 & 1 & 0 & 0 & | & 0 \\ 1 & 0 & 0 & 0 & | & 0 \end{bmatrix} \begin{bmatrix} I_{R_1} \\ I_{R_2} \\ V_{C_1} \\ V_{C_2} \\ \hline V_i \end{bmatrix} \quad (5.11)$$

$$\begin{bmatrix} I_{R_1} \\ I_{R_2} \\ V_{C_1} \\ V_{C_2} \end{bmatrix} = \begin{bmatrix} \cancel{I_{R_1}} & & & \\ & \cancel{I_{R_2}} & & \\ & & \underline{0} & \\ & & & \\ & & \cancel{I_{C_1}s} & \\ & \underline{0} & & \\ & & & \cancel{I_{C_2}s} \end{bmatrix} \begin{bmatrix} V_{R_1} \\ V_{R_2} \\ I_{C_1} \\ I_{C_2} \end{bmatrix} \quad (5.12)$$

TABLE 5.11
INPUTS AND OUTPUTS (Example 5.6)

<u>Input</u>			<u>Outputs</u>		
Number Designations	Type Designations	Cost	Number Designations	Type Designations	Cost
1	V_i	1	1	I_i	2
			2	I_o	2

TABLE 5.12
RESULTS (Example 5.6)

Case	No. of Inputs Chosen	No. of Outputs Chosen	No. Designation of Inputs Chosen	No. Designation of Outputs Chosen	Cost	Delta
1	1	1	1	1	3	0
2	1	1	1	2	3	1
3	1	2	1	1,2	5	0

5.7 Six Component RC Ladder Network

The ladder is shown in Figure 5.10 and a set of connection equations is given by equations (5.13) and (5.14). The other data used is:

Nominal value of each component	= 1
Number of components	= 6
Number of inputs	= 1
Number of outputs	= 2
Number of potentially variable parameters	= 6
Number of frequencies used	= 15
Starting frequency	= 1
Step size	= 1

For the costs assigned to inputs and outputs as per Table 5.11 (same as Example 5.6) the results for the three cases are shown in Table 5.13.

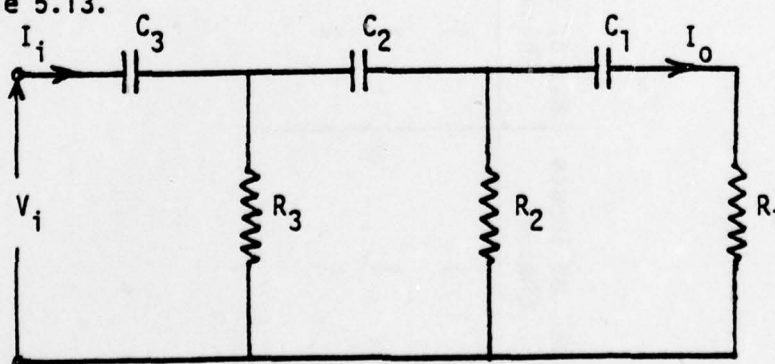


Figure 5.10 Six Component RC Ladder Network

$$\begin{bmatrix} V_{R_1} \\ V_{R_2} \\ V_{R_3} \\ I_{C_1} \\ I_{C_2} \\ I_{C_3} \\ \hline I_1 \\ I_0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & -1 & -1 & -1 & | & 1 \\ 0 & 0 & 0 & 0 & -1 & -1 & | & 1 \\ 0 & 0 & 0 & 0 & 0 & -1 & | & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & | & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & | & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & | & 0 \\ \hline 1 & 1 & 1 & 0 & 0 & 0 & | & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & | & 0 \end{bmatrix} \begin{bmatrix} I_{R_1} \\ I_{R_2} \\ I_{R_3} \\ V_{C_1} \\ V_{C_2} \\ V_{C_3} \\ \hline V_1 \end{bmatrix}$$

(5.13)

$$\begin{bmatrix} I_{R_1} \\ I_{R_2} \\ I_{R_3} \\ V_{C_1} \\ V_{C_2} \\ V_{C_3} \end{bmatrix} = \begin{bmatrix} \cancel{1/R_1} & & & & & \\ & \cancel{1/R_2} & & & & \\ & & \cancel{1/R_3} & & & \\ & & & \cancel{1/C_1 s} & & \\ & & & & \cancel{1/C_2 s} & \\ & & & & & \cancel{1/C_3 s} \end{bmatrix} \begin{bmatrix} V_{R_1} \\ V_{R_2} \\ V_{R_3} \\ I_{C_1} \\ I_{C_2} \\ I_{C_3} \end{bmatrix}$$

(5.14)

TABLE 5.13
RESULTS (Example 5.7)

Case	No. of Inputs Chosen	No. of Outputs Chosen	No. Designation of Inputs Chosen	No. Designation of Outputs Chosen	Cost	Delta
1	1	1	1	1	3	0
2	1	1	1	2	3	1
3	1	2	1	1,2	5	0

5.8 Eight Component RC Ladder Network

The ladder network is shown in Figure 5.11. Since the connection equations are a mere extension of those given by equations (5.10) and (5.11) and the cost data is the same as Table 5.11, the same will not be repeated here and in the next two examples. The other data used is:

Nominal value of each component	= 1
Number of components	= 8
Number of inputs	= 1
Number of outputs	= 2
Number of potentially variable parameters	= 8
Number of frequencies used	= 15

The results for two different choices of frequencies are shown in Table 5.14.

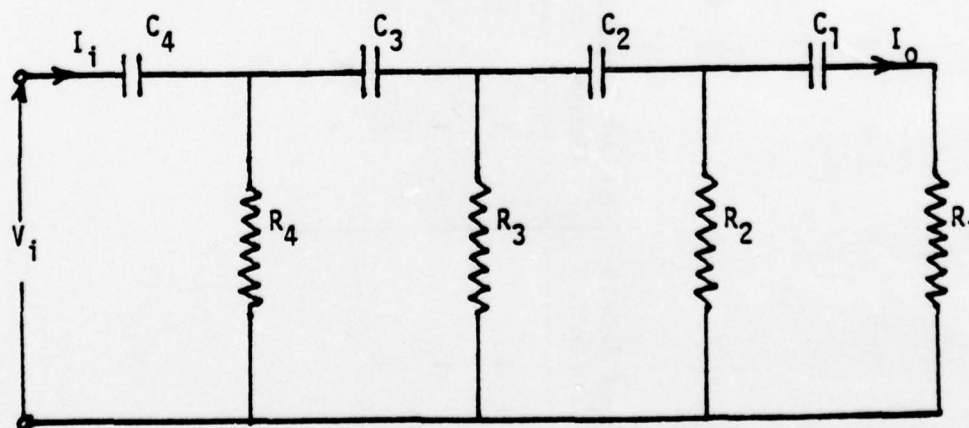


Figure 5.11 Eight Component RC Ladder Network

TABLE 5.14
RESULTS (Example 5.8)

Case	No. of Inputs Chosen	No. of Outputs Chosen	No. Designation of Inputs Chosen	No. Designation of Outputs Chosen	Cost	Delta	
						SFREQ=1 STEP=1	SFREQ=0.1 STEP=0.2
1	1	1	1	1	3	0	0
2	1	1	1	2	3	2	1
3	1	2	1	1,2	5	0	0

5.9 Ten Component RC Ladder Network

The ladder network is shown in Figure 5.12. The other data used is:

Nominal value of each component	= 1
Number of components	=10
Number of inputs	= 1
Number of outputs	= 2
Number of potentially variable parameters	=10
Number of frequencies used	=15

The results for four different choices of frequencies are shown in Table 5.15. It may be noted that even with a starting frequency of 0.1 and step size of 0.2, there is computer inaccuracy involved with single precision.

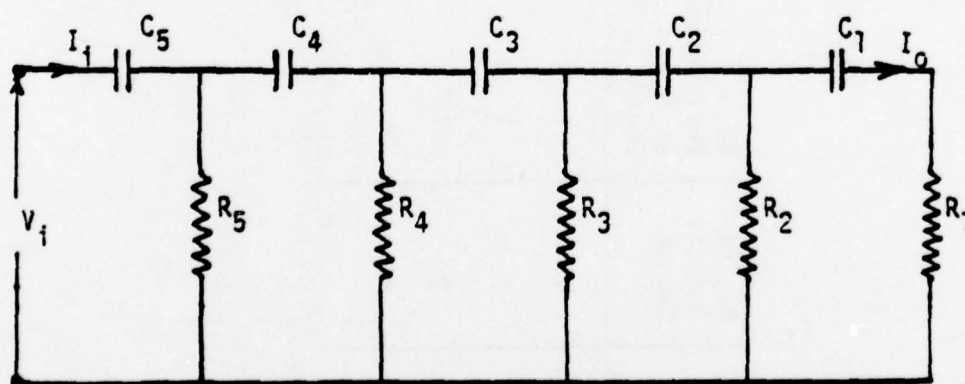


Figure 5.12 Ten Component RC Ladder Network

TABLE 5.15
RESULTS (Example 5.9)

Case	No. of Inputs Chosen	No. of Outputs Chosen	No. Designation of Inputs Chosen	No. Designation of Outputs Chosen	Cost	Delta			
						SFREQ=1 STEP=1	SFREQ= 0.5 STEP= 0.1	SFREQ= 0.5 STEP= 0.2	SFREQ= 0.1 STEP= 0.2
1	1	1	1	1	3	2	3	1	1
2	1	1	1	2	3	3	3	3	3
3	1	2	1	1,2	5	1	1	1	0

5.10 Twelve Component RC Network

The ladder network is shown in Figure 5.13. The other data used is:

Nominal value of each component	= 1
Number of components	=12
Number of inputs	= 1
Number of outputs	= 2
Number of potentially variable parameters	=12
Number of frequencies used	=15

The results for four different frequencies are shown in Table 5.16. It can be seen that with proper choice of frequencies, the results do improve and seem more realistic, yet a real need for double precision on the computer is felt.

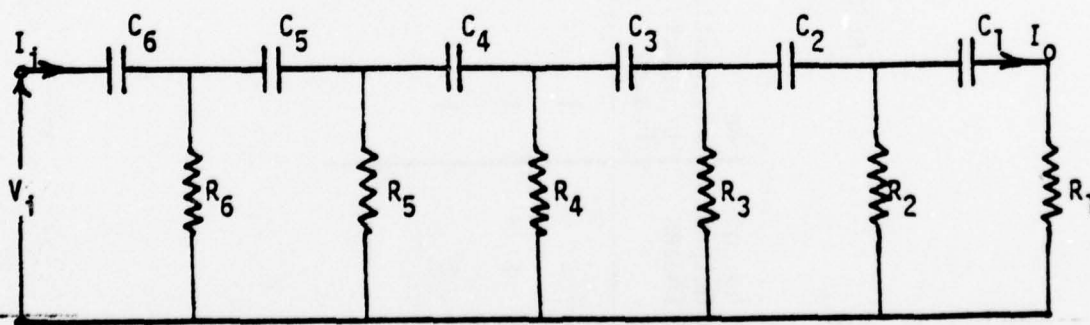


Figure 5.13 Twelve Component RC Ladder Network

TABLE 5.16
RESULTS (Example 5.10)

Case	No. of Inputs Chosen	No. of Outputs Chosen	No. Designation of Inputs Chosen	No. Designation of Outputs Chosen	Cost	Delta			
						SFREQ=1 STEP=1	SFREQ=0.5 STEP=0.1	SFREQ=0.5 STEP=0.2	SFREQ=0.1 STEP=0.2
1	1	1	1	1	3	3	3	2	2
2	1	1	1	2	3	4	3	4	5
3	1	2	1	1,2	5	0	1	1	0

CHAPTER VI

CONCLUSION

The purpose of this thesis has been to develop a measure of testability for analog circuits and systems and to illustrate its application to test point selection. This has been accomplished using the multifrequency fault analysis equations based on the component connection theory. Such a measure when formulated is an inherent property of the circuit or system and thus aids in test point allocation. The number and location of test points, depending on their electrically strategic location and accessibility in the system, is an important factor in testing and this serves as a major motivating force behind the development of a testability measure.

Computational aspects have been discussed and a precise approach to the computer evaluation of the measure of testability has been indicated. The software package developed by the author [17] is by no means the most efficient and professional package. Yet, at the same time, a family of examples drawn from various sections of circuit theory (active and passive components, filters, oscillators, and amplifiers, etc.), do show that the computer implementation has worked well for a variety of circuits. The examples also illustrate the intuition exercised in the choice of starting frequency, step size and the number of frequencies needed for analysis.

Not much work is done presently in the area of establishing such measures and their use for test point allocation for non-linear analog and digital circuits. A possible extension of this work may be in case of non-linear analog circuits with certain restrictions on non-linear-

ities involved, in order to achieve a global measure. There is a possibility of similar investigations in future for linear sequential circuits as well. On the computational front, there is a possibility of including an optimization algorithm to pick a "best" set of test points. This will have computational advantages for very large circuits and systems having a number of potentially variable circuit parameters and components.

REFERENCES

1. Rault, J.C., "A Bibliography on Fault Detection and Location in Digital Systems", Rpt. 74.404, Thomson-CSF, Inc., Paris, 1974. Also in Rational Fault Analysis, (ed. R. Saeks and S. R. Liberty), New York, Marcel Dekker, 1977.
2. Rault, J.C., "Fault Detection and Location in Analog Circuits-A Bibliography", IEEE Trans. on Circuit Theory, Vol. CT-19, pp. 634-640. Also in Rational Fault Analysis, (ed. R. Saeks and S. R. Liberty), New York, Marcel Dekker, 1977.
3. Friedman, A.D., and P.R. Menon, Fault Isolation in Digital Circuits, Englewood Cliffs, Prentice Hall, 1973.
4. Saeks, R., S.P. Singh and R.W. Liu, "Fault Isolation Via Components Simulation", IEEE Trans. on Circuit Theory, Vol. CT-19, 1972.
5. Ransom, M.N., and Saeks R., "The Connection Function-Theory and Application", Inter. Jour. of Circuit Theory and Application, Vol. 3, 1975.
6. Ransom, M.N., and Saeks, R., "A Functional Approach to Large Scale Systems", Proc. of the 10th Allerton Conf. on Circuit and System Theory, Univ. of Illinois, 1972.
7. Singh, S.P., "Structural Properties of Large-Scale Dynamical Systems", Ph.D. Thesis, Univ. of Notre-Dame, 1972.
8. Prasad, N. and J. Reiss, "The Digital Simulation of Inter-connected Systems", Proc. Conf. Inter. Assoc. of Cybernetics, Belgium, 1970.

9. Trauboth, H. and N. Prasad, "MARSYAS - A Software System for the Simulation of Digital System", Proc. Spring Joint Comp. Conf., 223-235, 1970.
10. Singh, S.P. and H. Trauboth, "MARSYAS", IEEE Circuits and Systems Society Newsletter, 7, 1973.
11. Ransom, M.N. and Saeks, R., "A Functional Approach to Fault Analysis", in Rational Fault Analysis, (ed. R. Saeks and S. R. Liberty), New York, Marcel Dekker, 1977.
12. Neudecker, H., "Some Theorems on Matrix Differentiation with Special Reference to Kronecker Matrix Products", Amer. Stat. Assoc. Journal., Sept. 1969, pp. 953-963.
13. Wilkinson, J.H., The Algebraic Eigenvalue Problem, Oxford, Clarendon Press, 1965.
14. Young, D.M., and R.T. Gregory, A Survey of Numerical Mathematics, Vol. I and Vol. II, Massachusetts, Addison Wesley, 1973.
15. Ortega, J., "The Givens-Householder Method for Symmetric Matrices", in Mathematical Methods for Digital Computers, by Ralston A., and H.S. Wilf, Vol. II, New York, Wiley, 1967.
16. Baldassari, M., Algebraic Varieties, Berlin, Springer Verlag, 1956.
17. Sen, N. and Saeks R., "ECMP - A Software Package for Computation of Measureability and Cost to Aid Test Point Selection", Texas Tech Univ., Lubbock, Texas, 1977.
18. Gray, P.E. and C.L. Searle, Electronic Principles - Physical Models and Circuits, New York, Wiley, 1969.

20. Rao, C.R., and Mitra, S.K., "Generalized Inverse of Matrices and Its Applications", New York, Wiley, 1971.
21. Fleming, W.H., "Functions of Several Variables", Reading, Addison Wesley, 1965.
22. Tassel, D.V., Program Style, Design, Efficiency, Debugging and Testing, Englewood Cliffs, Prentice Hall, 1974.

APPENDIX

SOME CONCEPTS FROM ALGEBRAIC GEOMETRY

Zero Set:

Let $f(x_1, x_2, x_3, \dots, x_n) = S$ be a polynomial in n variables with complex coefficients.

Their, the zero set of S is given by:

$$[(x_1, x_2, x_3, \dots, x_n) : f(x_1, x_2, x_3, \dots, x_n) = 0]$$

For such a polynomial in n variables, one can fix $(n - 1)$ variables and obtain the discrete set for the remaining variables. Then by varying the original $(n - 1)$ variables one can parameterize the entire zero set of the original polynomial using a maximum of $(n - 1)$ variables. As such, the zero set of a polynomial in n variables is, indeed $(n - 1)$ dimensional.

Algebraic Hypersurface:

V is an algebraic hypersurface if and only if

$$V \subset \mathbb{R}^n$$

$$V \neq \emptyset$$

and V is the zero set of the polynomial

$$f(x_1, x_2, \dots, x_n).$$

Algebraic Variety:

An algebraic variety is the intersection of finitely many hypersurfaces. i.e., an algebraic variety is the zero set of a set of simultaneous polynomial equations in several variables.

Almost All:

"Almost all" points of an algebraic variety V are said to have a

given property α if the points of V which do not have the property α belong to a proper algebraic subset of V . [16] This means that, for an algebraic variety for a polynomial in n variables, the $(n - 1)$ dimensional subset of n dimensional subspace is small. Formally, the relative Lebesgue measure of such a set is zero thus showing that the above definition of "almost all" is consistent with the classical measure theoretic approach.

A SEARCH ALGORITHM FOR THE SOLUTION OF THE FAULT
DIAGNOSIS EQUATIONS*

H.M.S. Chen**

*This research supported in part by Office of Naval Research Contracts 75-C-0924 and 76-C-1136.

**Presently with National Semiconductor Corporation, San Jose, Calif.

TABLE OF CONTENTS

	Page
ACKNOWLEDGMENTS	ii
LIST OF FIGURES	iii
LIST OF TABLES	v
I. INTRODUCTION	1
The Component Connection Model	1
The Fault Diagnosis Equations	3
Overview of the Thesis	6
II. SOLUTION OF THE FAULT DIAGNOSIS EQUATIONS-LINEAR CASE	7
Introduction to the Linear Problem	7
Exterior Product	7
Computing Algorithms and a Modified Exterior Product	8
Error Analysis	15
III. SOLUTION OF THE FAULT DIAGNOSIS EQUATIONS-NON- LINEAR CASE	18
Techniques for Evaluating the Connection Function Using Householder's Formula	18
Search Algorithms for the Solution of the Fault Diagnosis Equations	20
Examples and Computer Simulation	22
IV. JACOBIAN TECHNIQUE SOLVING FOR THE FAULT DIAGNOSIS EQUATIONS	33
V. CONCLUSION	40
REFERENCES	42

LIST OF FIGURES

Figure		Page
1.	Development of the component connection model. .	4
2.	Butterworth filter	23
3.	One stage transistor amplifier	28
4.	Amplifier equivalent circuit	28

LIST OF TABLES

TABLE	Page
1. Computer Simulation Results for the Butterworth Filter.	26
2. Computer Simulation Results for one Stage Transistor Amplifier.	30
3. Computer Simulation Results for the 4th order Butterworth Filter.	36
4. Computer Simulation results for the 8th order Butterworth Filter.	38

CHAPTER I

INTRODUCTION

Historically investigations in the area of fault diagnosis have been conducted along two independent paths. Diagnosis techniques for digital systems are combinatorial in nature, faulty components being detected by observing the system response to a set of input test signals. In an analog system, fault diagnosis techniques are based on the dynamical properties of the components, the faulty elements being located by measuring the system gain at various frequencies. Bibliographies in these two areas have been compiled by Rault.^{1,2} A quick review of the literature reveals that research on the fault diagnosis of analog systems has been left behind by that in the digital case. Although many achievements have been made in digital fault diagnosis, the rapid development of modern integrated circuit technology also demands automated analog fault diagnosis. It is believed that automated fault diagnosis of analog systems will prove to be very important in production line quality control as well as in maintenance and service.

The Component Connection Model

The classical expression for the response of a system is in terms of the so called transfer function

REPRODUCTION PAGE NOT FILLED
BLANK

and its inputs. This representation, however, becomes too complex to analyse in the case of large scale systems. A new approach based on separating the components and the connectivity of the system has been developed to meet this need.³ The component connection model for an interconnected dynamical system carries a complete description of the system in the form of two separate equations; a decoupled dynamical equation characterizing the components and a coupled algebraic equation characterizing the connections. Figures 1(a) and 1(b) illustrate the development of the component connection model. In figure 1(a), the overall system is characterized by the equation

$$y = Su \quad (1.1)$$

where the operator S may be interpreted as a transfer function or a set of linear or nonlinear state equations; while the components can be modeled using the component input and output variables a and b via

$$b = Za \quad (1.2)$$

Pictorially, the shaded area of figure 1(a) represents the components and the hatched area represents the connections. In figure 1(b), the components and connections of the system are separated which gives us a "donut shaped" connection box whose inputs are u and b and whose outputs

are y and a . Now, the connections are characterized entirely by linear algebraic constraints, hence it is reasonable to model them by the following matrix equation.

$$\begin{bmatrix} a \\ y \end{bmatrix} = \begin{bmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} b \\ u \end{bmatrix} \quad (1.3)$$

Equations (1.2) and (1.3) are termed the component connection model for the system. The applications of this model include sensitivity analysis, computer aided design, and fault diagnosis. In the sequel, the third application will be discussed in depth.

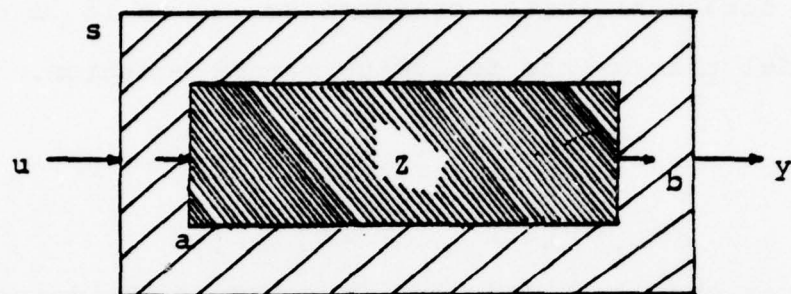
The Fault Diagnosis Equations

Combining (1.2) and (1.3) one can obtain the relation

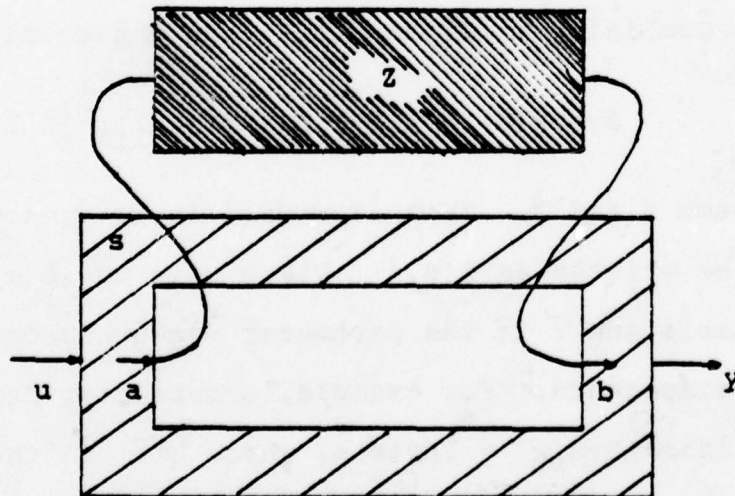
$$S = f(Z) = L_{22} + L_{21} (I - ZL_{11})^{-1} Z L_{12} \quad (1.4)$$

between S and Z . When expressed in frequency domain, Z can be written as $Z(s, \bar{r})$, where s is the complex frequency variable and \bar{r} is the parameter vector associated with the components. For example, a capacitor has the representation $Z(s, \bar{r}) = 1/c(\bar{r})s$, where $c(\bar{r})$ is the capacitance. Thus we can rewrite (1.4) as

$$S(s, \bar{r}) = f(Z(s, \bar{r})) = L_{22} + L_{21} (I - Z(s, \bar{r})L_{11})^{-1} Z(s, \bar{r})L_{12} \quad (1.5)$$



(a)



(b)

Figure 1. Development of the component connection model.

Equation (1.5) is called the fault diagnosis equation. The diagnosis problem is then to determine all the solutions, \bar{r} , to this equation given S and the connection matrices. Unfortunately the solvability of the equation depends on the left invertibility of the matrix:

$$k = L_{12}^T \otimes L_{12} \quad (1.6)$$

where \otimes denotes the Kronecker matrix product.⁴ This, in turn, implies that the system must have a large number of test points. In practice it is not always possible to obtain access to as many test points as needed. The other alternative is to take measurements of S at several frequencies, this yields more equations without increasing the number of unknowns and renders the equation solvable. That is, one has to solve a set of simultaneous equations expressed in vector form by:

$$\begin{bmatrix} S(s_1, \bar{r}) \\ S(s_2, \bar{r}) \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ S(s_n, \bar{r}) \end{bmatrix} = \begin{bmatrix} f(Z(s_1, \bar{r})) \\ f(Z(s_2, \bar{r})) \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ f(Z(s_n, \bar{r})) \end{bmatrix} \quad (1.7)$$

This approach is based on the assumption that component variations with frequency are known and are non-faulty and that the connections of the system remain fixed. Furthermore, in a large scale system usually only a very small percentage of components will fail and the failed components are usually confined to certain known "burst" groups. Exploiting the above special properties, the fault diagnosis equations can be solved efficiently.

Overview of the Thesis

In this thesis, several search algorithms for the solutions of the fault diagnosis equations will be presented. As a preliminary, some methods for solving a system of linear equations will be discussed in Chapter II. With the motivation of the linear problem, we develop a talk about the search algorithm for the solution of the fault diagnosis equations in Chapter III. This approach takes the advantage of Householder's formula.⁵ Finally, a method using the Jacobian matrix of the fault diagnosis equations will be investigated. This method, although not satisfactory in all cases, gives us some highlights for further research.

CHAPTER II

SOLUTION OF THE FAULT DIAGNOSIS EQUATIONS-LINEAR CASE

Introduction to the Linear Problem:

Before considering the nonlinear fault diagnosis equations, let us investigate the linear problem. We consider a system of linear equations with the special property that the solution has only one (or several) nonzero coordinates while the other coordinates are all identically zero. We want to find the coordinates where the nonzero elements are located. In other words, we have the system of linear equations.

$$\sum_{i=1}^n \bar{a}_i x_i = \bar{y} = [\bar{a}_1 \bar{a}_2 \dots \bar{a}_n] \bar{x}$$

where \bar{a}_i , \bar{x} , \bar{y} are all n -tuples and we assume that at most p ($p \ll n$) of the entries in \bar{x} are nonzero. If $x_i \neq 0$ and $x_j \Big|_{j \neq i} = 0$, then the equation reduces to $\bar{a}_i x_i = \bar{y}$. This implies that \bar{a}_i and \bar{y} are linearly dependent. By virtue of the characteristics of the exterior product, one can thus solve the problem efficiently.

Exterior Product

It is well known in multilinear algebra that a set of vectors $\bar{a}_1, \bar{a}_2, \dots, \bar{a}_n$ in a vector space V is linearly

dependent if and only if the exterior product of the vectors is zero. We can apply this property of exterior product (\wedge) to solve the aforementioned linear problem.

Suppose there are P vectors in a n dimensional Euclidean space, $\bar{a}_1, \bar{a}_2, \dots, \bar{a}_p$, with $p \leq n$, then the exterior product of the p n -tuples is a vector of $c(n, n - p) = \frac{n!}{(n-p)!p!}$ entries. For detailed discussions please refer to reference 6. As an example, let

$$\bar{a}_1 = [x_1 x_2 x_3]' \text{ and } \bar{a}_2 = [y_1 y_2 y_3]'$$

then the exterior product of \bar{a}_1 and \bar{a}_2 is

$$\bar{a}_1 \wedge \bar{a}_2 = [x_1 y_2 - y_1 x_2 \quad x_1 y_3 - y_1 x_3 \quad x_2 y_3 - y_2 x_3]'$$

Here, the number of entries of $\bar{a}_1 \wedge \bar{a}_2$ is

$$\frac{3!}{(3-2)!2!} = 3$$

Computing Algorithms and a Modified Exterior Product

Our purpose is to establish a computational algorithm that will determine if \bar{y} is a linear combination of a group of $p-1$ vectors. This may, in turn, be used to locate the nonzero entries of X -vector. Since we are interested in on-line fault analysis, the information from the coefficient matrix which is known a-priori, will be exploited first, while the use of the measured response vector (\bar{y}) is kept

to a minimum. Furthermore, by modifying the computing procedure of the usual exterior product, one can solve the problem efficiently.

Algorithms:

A. $P = 2$

Given $\bar{y} = [y_1 y_2 \dots y_n]$ and $\bar{a} = [a_1 a_2 \dots a_n]$, we want to determine if \bar{y} can be a multiple of \bar{a} . For example, consider two vectors $\bar{y} = [1243]'$ and $\bar{a} = [2486]'$. Taking the exterior product of \bar{y} and \bar{a} , we have

$$\bar{y} \wedge \bar{a} = [0 \ 0 \ 0 \ 0 \ 0 \ 0]'$$

This implies that \bar{y} and \bar{a} are linearly dependent, thus \bar{y} is a multiple of \bar{a} . Note that the evaluation of six 2×2 minors is involved in the computation. In fact, for the purpose of determining whether \bar{y} is a multiple of \bar{a} only three sequential 2×2 minors need to be computed if there are no zero entries contained in \bar{a} . In addition, if any 2×2 sequential minor is not zero during the computation, it follows immediately that \bar{y} can not be a multiple of \bar{a} . To prove the statement, assume the first two sequential 2×2 minors are zero, we have

$$\alpha_1 \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} + \beta_1 \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = 0 \Rightarrow \alpha_1 y_2 + \beta_1 a_2 = 0 \quad (2.1)$$

$$\alpha_2 \begin{bmatrix} y_2 \\ y_2 \end{bmatrix} + \beta_2 \begin{bmatrix} a_2 \\ a_3 \end{bmatrix} = 0 \Rightarrow \alpha_2 y_2 + \beta_2 a_2 = 0 \quad (2.2)$$

In (2.1) and (2.2), α_1 and α_2 must be nonzero, otherwise it will contradict the fact that \bar{a} has all nonzero entries. After normalizing α_1 and α_2 , we obtain

$$y_2 = e_1 a_2 = e_2 a_2 \quad (2.3)$$

thus $e_1 = e_2$ and

$$\begin{bmatrix} y_1 \\ y_2 \\ y_2 \end{bmatrix} = e_1 \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

Applying the same argument inductively, we have $e_1 = e_2 = \dots = e_{n-1}$, that is, \bar{y} is a multiple of \bar{a} . Next, consider the case where \bar{a} contains some zero entries. Let $\bar{a} = [1 \ 0 \ 2]'$ and $\bar{y} = [1 \ 3 \ 2]'$. It is clear that \bar{y} cannot be a multiple of \bar{a} since the second coordinate of \bar{a} is zero while that of \bar{y} is not. In order for \bar{y} to be a multiple of \bar{a} , it is necessary that for each zero entry of \bar{a} , the corresponding entry of \bar{y} should also be zero. Suppose $\bar{a} = [1 \ 0 \ 2]'$ and $\bar{y} = [2 \ 0 \ 4]'$, then \bar{y} is a multiple of \bar{a} if and only if $\bar{y}' = [2 \ 4]'$ is a multiple of $\bar{a}' = [1 \ 2]$. This fact suggests that we can consider the simpler case of \bar{y}' and \bar{a}' rather than \bar{y} and \bar{a} . Motivated by discussions

above, the modified exterior product algorithm for the $p = 2$ case follows:

(1) If $a_i = 0$, do the following tests; otherwise go to (2).

(i) If the corresponding $y_i = 0$, delete the i^{th} row and go to (1); otherwise, stop. (\bar{y} cannot be a multiple of \bar{a} .)

(2) Finally we reach the situation where every entry of \bar{a} is nonzero. If at least one of the sequential 2×2 minors of $[\bar{y}' \ \bar{a}']$ is not zero, stop (\bar{y} cannot be a multiple of \bar{a}); otherwise, \bar{y} is a multiple of \bar{a} .

B. $P = 3$.

Given \bar{y} , a response vector, \bar{a} and \bar{b} , two linearly independent vectors, the objective is to determine if \bar{y} can be a linear combination of \bar{a} and \bar{b} . As an example, let $\bar{y} = [1 \ 2 \ 4 \ 6 \ 5]'$, $\bar{a} = [2 \ 4 \ 8 \ 12 \ 10]'$, and $\bar{b} = [3 \ 6 \ 12 \ 18 \ 15]'$. The exterior product of the three vectors is $\bar{y} \wedge \bar{a} \wedge \bar{b} = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]'$. It follows that \bar{y} , \bar{a} and \bar{b} are linearly dependent, which in turn implies that \bar{y} is a linear combination of \bar{a} and \bar{b} . Note there are ten 3×3 minors calculations involved. If we further assume that no sequential 2×2 minor of $[\bar{a} \ \bar{b}]$ is zero, evaluation of three 3×3 sequential minors is sufficient to determine whether \bar{y} is a linear combination of \bar{a} and

5. If any of the three 3×3 sequential minors is not zero, it follows immediately that \bar{y} cannot be a linear combination of \bar{a} and \bar{b} . The proof of the above statement can be developed in a similar manner as that for the $p = 2$ case.

Next, the case where some of the sequential 2×2 minors of $[\bar{a} \ \bar{b}]$ are zero will be investigated. Let $\bar{y} = [1 \ 3 \ 4 \ 5]'$, $\bar{a} = [0 \ 6 \ 8 \ 10]'$, and $\bar{b} = [0 \ 9 \ 16 \ 15]'$, it follows that the first 2×2 minor of $[\bar{a} \ \bar{b}]$ is zero. Since $y_1 = 1 \neq 0$, it is clear that \bar{y} cannot be a linear combination of \bar{a} and \bar{b} . If \bar{y} is changed to $\bar{y} = [0 \ 3 \ 4 \ 5]'$, then \bar{y} can be a linear combination of \bar{a} and \bar{b} if and only if $\bar{y}' = [3 \ 4 \ 5]'$ can be a linear combination of $\bar{a}' = [6 \ 8 \ 10]'$ and $\bar{b}' = [9 \ 16 \ 15]'$. Thus the dimension of the original problem has been decreased by one. As in the last example, consider the case where $\bar{y} = [1 \ 3 \ 4 \ 5]'$, $\bar{a} = [2 \ 3 \ 8 \ 10]'$ and $\bar{b} = [4 \ 6 \ 16 \ 15]'$. The first 2×2 minor of $[\bar{a} \ \bar{b}]$ is zero, while that of $[\bar{y} \ \bar{a}]$ is not zero, it is obvious that \bar{y} cannot be a linear combination of \bar{a} and \bar{b} . To sum up the above discussions, we have the following algorithm:

- (1) Precompute every sequential 2×2 minor of $[\bar{a} \ \bar{b}]$.
- (2) If the i^{th} minor is equal to zero, do the following tests; otherwise go to (4).

$$(i) \text{ Assume } \begin{vmatrix} a_i & b_i \\ a_{i+1} & b_{i+1} \end{vmatrix} = 0$$

If $a_i = b_i = 0$, go to (ii); otherwise, go to (iii).

Do a similar test for a_{i+1} and b_{i+1} .

(ii) If the corresponding $y_i = 0$, cross the i^{th} row out and go to (2); otherwise, stop. (\bar{y} cannot be a linear combination of \bar{a} and \bar{b} .)

(III) If $[a_i a_{i+1}]' = 0$ or $[b_i b_{i+1}]' = 0$, mark the position.

(iv) If the determinant of $[y_i y_{i+1}]'$ with any non-zero $[a_i a_{i+1}]'$ or $[b_i b_{i+1}]'$ is zero, cross the i^{th} row out and go to (2); otherwise, stop. (\bar{y} cannot be a linear combination of \bar{a} and \bar{b} .)

(3) After crossing out any row, we must perform a new minor test and proceed as in (2).

(4) At last, the situation is reduced to that for $[\bar{a}' \bar{b}']$ where no sequential 2×2 minor is zero. If at least one of the 3×3 sequential minors of $[\bar{a}' \bar{b}' \bar{y}']$ is nonzero, stop. (\bar{y} cannot be a linear combination of \bar{a} and \bar{b}); otherwise, \bar{y} is a linear combination of \bar{a} and \bar{b} .

C. General Case

We have \bar{y} , a response vector, $\bar{a}_1, \bar{a}_2, \dots, \bar{a}_{p-1}$, $p-1$ linearly independent vectors. The objective is to determine if \bar{y} can be a linear combination of the $p-1$ vectors.

(1) Precompute every $(p-1) \times (p-1)$ minor of $[\bar{a}_1, \bar{a}_2, \dots, \bar{a}_{p-1}]$.

- (2) If any i^{th} minor is zero, do the following tests; otherwise, go to (4)
- (i) If $a_{1,i} = a_{2,i} = \dots = a_{p-1,i} = 0$, go to (ii); otherwise, go to (iii).
 - (ii) If the corresponding $y_i = 0$, cross the i^{th} row out and go to (3); otherwise, stop. (\bar{y} cannot be a linear combination of the $p-1$ vectors)
 - (iii) If any of $[a_{j,i} \ a_{j,i+1} \ a_{j,i+p-2}]'$ ($j = 1, 2, \dots, p-1$) is zero, mark the position and go to (iv). If two or more of them are zero vectors, go to (v).
 - (iv) If the determinant of $[y_i, y_{i+1}, \dots, y_{i+p-2}]'$ with any remaining $p-2$ nonzero vectors is zero, go to (v); otherwise, stop. (\bar{y} cannot be the linear combination of the $p-1$ vectors.)
 - (v) Proceed as in $p-1$ case, determine which one of the $p-1$ rows is a linear combination of the other $p-2$ ones and cross the located row out, then go to (3).
- (3) After crossing the row out, we must do $(p-2)$ additional minor tests.
- (4) Finally, the situation is reduced to the case where no $(p-1) \times (p-1)$ sequential minor of $[\bar{a}'_1 \ \bar{a}'_2 \ \dots \ \bar{a}'_{p-1}]$ is zero.
- If at least one $p \times p$ sequential minor of $[\bar{y}' \ \bar{a}'_1 \ \bar{a}'_2 \ \dots \ \bar{a}'_{p-1}]$ is nonzero, stop; (\bar{y} cannot be a linear combina-

tion of the $p-1$ vectors) otherwise, \bar{y} is a linear combination of the $p-1$ vectors. From the above discussions, we can draw an important conclusion: Once some $(p-1) \times (p-1)$ minor is zero, we either stop or cross the i^{th} row out. Thus computing time is reduced. Also note that in a situation where the $p-1$ vectors a_1, a_2, \dots, a_{p-1} are known a-priori (such as might arise in the solution of the linearized fault diagnosis equations) much of the work with $p-1 \times p-1$ minors can be done off-line with only the final computation of $p \times p$ minors (via a Laplace expansion into procomputed $p-1 \times p-1$ minors) being done on-line.

D. Computer Simulation and Results (for $p = 1$ Case)

Using the above described algorithm a 10th order system was simulated on the CDC-1604 computer. The resultant number of multiplications proved to be relative low compared to that required for the inversion of a 10×10 matrix.* Thus, the exploitation of the special property associated with special form of the solution of the fault diagnosis equations really saves computer time, implying the possibility of on-line fault diagnosis.

Error Analysis

Instead of restricting the components of \bar{x} to be zero, we allow for the possibility that the zero entries

* For the case of one non-zero coordinate in \bar{x} , we estimate that about $(2.5) n$ multiplications are required.

are corrupted by noise, ϵ_i . Now, an error analysis formula for the $p = 1$ case will be derived. Let C_i be the nonzero component, the maximum absolute value of ϵ_i will be ϵ_M , and " $\overset{\circ}{\wedge}$ " denotes the modified exterior product of sequential minors. The problem to be solved becomes

$$\bar{y} = \sum_{j \neq i}^n \epsilon_j \bar{a}_j + C_i \bar{a}_i.$$

Taking the modified exterior product of \bar{y} and \bar{a}_k , we have

$$\bar{y} \overset{\circ}{\wedge} \bar{a}_k = C_i \bar{a}_i \overset{\circ}{\wedge} \bar{a}_k + \sum_{j \neq i, k}^n \epsilon_j \bar{a}_j \overset{\circ}{\wedge} \bar{a}_k = L_k$$

Let $k = i$, since $\bar{a}_i \overset{\circ}{\wedge} \bar{a}_i = 0$, then

$$S = \sum_{j \neq i}^n \epsilon_j \bar{a}_j \overset{\circ}{\wedge} \bar{a}_k \leq \epsilon_M \sum_{j \neq i}^n |\bar{a}_j \overset{\circ}{\wedge} \bar{a}_k| = S_1$$

Applying the triangle inequality, $|A - B| \geq |A| - |B|$, to L_k , yields

$$\begin{aligned} |L_k| &\geq |C_i \bar{a}_i \overset{\circ}{\wedge} \bar{a}_k| - \left| \sum_{j \neq i, k}^n \epsilon_j \bar{a}_j \overset{\circ}{\wedge} \bar{a}_k \right| \\ &\geq |C_i| |\bar{a}_i \overset{\circ}{\wedge} \bar{a}_k| - \epsilon_M \sum_{j \neq i, k}^n |\bar{a}_j \overset{\circ}{\wedge} \bar{a}_k| = L_{k1} \end{aligned}$$

In order to locate where the non-zero coordinate, we argue that $S_1 < L_{k1}$, then

$$\epsilon_M \sum_{j \neq i}^n |a_j \overset{\circ}{\wedge} a_i| < |c_i| |\bar{a}_i \overset{\circ}{\wedge} \bar{a}_k| - \epsilon_M \sum_{j \neq i, k}^n |\bar{a}_j \overset{\circ}{\wedge} \bar{a}_k|$$

which implies that

$$\epsilon_M \left(\sum_{j \neq i}^n |\bar{a}_j \overset{\circ}{\wedge} \bar{a}_i| + \sum_{j \neq i, k}^n |\bar{a}_j \overset{\circ}{\wedge} \bar{a}_k| \right) < |c_i| |\bar{a}_i \overset{\circ}{\wedge} \bar{a}_k|$$

That is,

$$\frac{\epsilon_M}{|c_i|} < \frac{|\bar{a}_i \overset{\circ}{\wedge} \bar{a}_k|}{\sum_{j \neq i}^n |\bar{a}_j \overset{\circ}{\wedge} \bar{a}_i| + \sum_{j \neq i, k}^n |\bar{a}_j \overset{\circ}{\wedge} \bar{a}_k|}$$

This condition thus represents a bound on the ratio of the noise in the "zero" entries of \bar{x} to the amplitude of the nonzero entry which will assure the validity of our exterior product algorithm.

CHAPTER III

SOLUTION OF THE FAULT DIAGNOSIS EQUATIONS-NONLINEAR CASE

As mentioned in Chapter I, the multifrequency analysis gives us a way in locating the faults of a system without access to many test points. The idea behind the multifrequency diagnosis is based on the dynamical properties of the components and the assumption that the variations in the components from nominal status are due entirely to changes in a fixed parameter vector, \bar{r} , with the dynamical character of Z unchanged. If one further assumes that only one component or one "burst" group of components has failed, the problem can be numerically solved very economically by virtue of Householder's formula.

Techniques for Evaluating the Connection Function Using Householder's Formula

Let \hat{A} , A , C , and R denote matrices of order $n \times n$, $n \times n$, $n \times k$, and $k \times n$, respectively, and let

$$\hat{A} = A + CR \quad (3.1)$$

If we know A^{-1} in advance, and k is "small", we can evaluate \hat{A}^{-1} by Householder's formula:

$$\hat{A}^{-1} = [I_n - A^{-1}C(I_k + RA^{-1}C)^{-1}R]A^{-1} \quad (3.2)$$

Not that only the inversion of a $k \times k$ matrix is involved in the computation of \hat{A}^{-1} . Now assume that one entry (or a group of entries) of the composite component transfer function matrix, $Z(s, \bar{r})$, has changed, then the new matrix becomes

$$Z_1(s, \bar{r}) = Z(s, \bar{r}) + \begin{bmatrix} \Delta & 0 \\ 0 & 0 \end{bmatrix} \quad (3.3)$$

where Δ is the 1×1 (or $k \times k$) matrix of entries in $Z(s, \bar{r})$ which are affected by the parameter change.*

By (3.3), we have

$$\begin{aligned} I_n - Z_1(s, \bar{r})L_{11} &= (I_n - Z(s, \bar{r})L_{11}) - \begin{bmatrix} \Delta & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} L_{11}' \\ L_{11}^2 \end{bmatrix} \\ &= (I_n - Z(s, \bar{r})L_{11}) - \begin{bmatrix} \Delta \\ 0 \end{bmatrix} [L_{11}'] \end{aligned} \quad (3.4)$$

where L_{11}' is a $1 \times n$ (or $k \times n$) matrix. Applying Householder's formula, we have

$$\begin{aligned} [I_n - Z_1(s, \bar{r})L_{11}]^{-1} &= [I_n + (I_n - Z(s, \bar{r})L_{11})^{-1} \begin{bmatrix} \Delta \\ 0 \end{bmatrix}] \\ &= \{I_k - L_{11}'(I_n - Z(s, \bar{r})L_{11})^{-1} \begin{bmatrix} \Delta \\ 0 \end{bmatrix}\}^{-1} L_{11}' (I_n - Z(s, \bar{r})L_{11})^{-1} \end{aligned} \quad (3.5)$$

Substituting equation (3.5) into equation (1.5), we obtain the new connection function. Thus, we can express

*Since $Z(s, \bar{r})$ is block diagonal, k is typically very nearly equal to the number of parameters which have changed.

the connection function as a function of Δ , $S_1(\bar{r}, \Delta)$, once the non-variable components of the parameter vector, \bar{r} , are fixed at their nominal values. The advantage of such a technique is that in the process of evaluating the new connection function S_1 , there is essentially no matrix inversion involved. (or just the inversion of a $k \times k$ matrix when Δ is k by k .)

Search Algorithms for the Solution of the Fault Diagnosis Equations

One application of the optimization theory is the equation solving by search. Given a set of equations

$$g_i(\bar{X}) = 0 \quad i = 1, 2, \dots, n \quad (3.6)$$

The objective is to find the solution vector $\bar{x} = [x_1 x_2 \dots x_n]'$ which satisfy (3.6). To form an equivalent minimization problem, a performance measure

$$J(\bar{X}) \triangleq \sum_{i=1}^n [g_i(\bar{X})]^2 \quad (3.7)$$

is defined.

Note that $J(\bar{x})$ is always non-negative, therefore, if an \bar{x} exists for which $J(\bar{x}) = 0$, it must be a minimum point for $J(\bar{x})$ and a solution of (3.6). Thus, the solutions to (3.6) are obtained by searching for the minima of $J(\bar{x})$.

A very efficient algorithm for finding the minimum of a convex function of one variable is the Golden section search. In the following we will describe the algorithm. The objective here is to locate the minimum of $f(t)$, $f(t')$, between $[0, \infty]$. Two positive numbers c and s are supplied; also, the Fibonacci numbers, $F_1 = (3 - \sqrt{5})/2 \approx 0.38$ and $F_2 = (\sqrt{5} - 1)/2 \approx 0.68$ will be employed.

- (1) Compute $f(c)$ and $f(0)$.
- (2) If $f(c) \geq f(0)$, set $l_0 = 0$, $m_0 = c$, and go to (7); otherwise, go to (3).
- (3) Set $i = 0$, and $d_0 = 0$.
- (4) Set $d_{i+1} = d_i + c$.
- (5) Compute $f(d_{i+1})$.
- (6) If $f(d_{i+1}) \geq f(d_i)$, set $l_0 = d_{i-1}$, $m_0 = d_{i+1}$, and go to (7); otherwise, set $i = i+1$ and go to (4).

At this point in the algorithm, $t' \in [l_0, m_0]$, and we want to reduce the interval containing t' .

- (7) Set $j = 0$.
- (8) Set $r_j = (m_j - l_j)$.
- (9) If $r_j \leq s$, go to (12); otherwise, go to (10).
- (10) Set $a_j = l_j + F_1 r_j$ and $b_j = l_j + F_2 r_j$.
- (11) If $f(a_j) < f(b_j)$, set $l_{j+1} = l_j$, $m_{j+1} = b_j$, set $j = j + 1$ and go to (8); otherwise, set $l_{j+1} = a_j$, $m_{j+1} = m_j$, set $j = j + 1$ and go to (8).
- (12) Set $t' = (l_j + m_j)/2$ and stop.

Suppose for the fault diagnosis problem there is only one component parameter r_i which changes its value. To locate r_i , the function to be minimized is

$$f(t) = ||S_1(s, t) - S(s)||^2$$

where $t = \Delta_i + r_i$ denotes the failed value of the component. The minimum point t' for the changed component should represent the exact value and the minimum value at the point should approach zero. Here, $S(s)$ denotes the transfer function matrix for the faulty system. If there is a group of parameters which change simultaneously, the function to be minimized becomes

$$f(\bar{v}) = ||S_1(s, \bar{v}) - S(s)||^2$$

where \bar{v} is a column vector, the dimension of the vector depends on how many components are assumed to have failed at once. In this case, the Golden section search method is not applicable, and some other "decent algorithm" like the methods of steepest descent and conjugate gradient must be used. The minimum point \bar{v}' should represent the exact failed values of the group of components and the minimum value at that point is expected to approach zero.

Examples and Computer Simulation

As a first example, consider the Butterworth filter

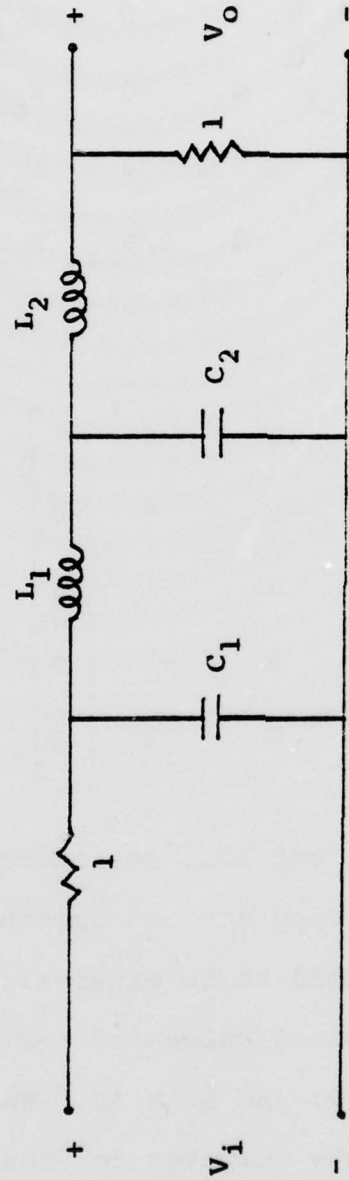


Figure 2. Butterworth filter.

shown in figure 2.

For which the component connection model takes the form

$$\begin{bmatrix} V_{c1} \\ i_{L1} \\ V_{c2} \\ i_{L2} \end{bmatrix} = \begin{bmatrix} 1/S_{c1} & 0 & 0 & 0 \\ 0 & 1/S_{L1} & 0 & 0 \\ 0 & 0 & 1/S_{c2} & 0 \\ 0 & 0 & 0 & 1/S_{L2} \end{bmatrix} \begin{bmatrix} i_{c1} \\ V_{L1} \\ i_{c2} \\ V_{L2} \end{bmatrix} \quad (3.8)$$

and

$$\begin{bmatrix} i_{c1} \\ V_{L1} \\ i_{c2} \\ V_{L2} \\ \hline V_o \end{bmatrix} = \begin{bmatrix} -1 & -1 & 0 & 0 & 1 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} V_{c1} \\ i_{L1} \\ V_{c2} \\ i_{L2} \\ \hline V_i \end{bmatrix} \quad (3.9)$$

Here we imbed the source and load resistors into the connection structure since they are not components in the usual sense and are assumed to be external to the circuit under test. Let the nominal values of the parameters be $C_1 = 10$, $L_1 = 20$, $C_2 = 30$, and $L_2 = 40$. We then simulated the system on the CDC-1604 computer for four sets of parameters with one component changed at a time. The algorithm used is Golden section search method and the function to

be minimized is

$$f(t) = ||S_1(s,t) - S(s)||^2$$

where t denotes the new value of each component. The results are summarized in Table 1.

From table 1 it is observed that for the failed component, $f(t')$ is very close to zero and t' is exactly the new component value. Thus the fault can be detected and isolated. This search technique proves to be very efficient for locating one faulty component of a system and we believe that the results are very significant in the area of the fault diagnosis work.

As a more sophisticated example, let us look at a one stage transistor amplifier as shown in figure 3. The hybrid equivalent circuit of the amplifier is shown in figure 4.⁸ The component connection model becomes³ as shown in (3.10) and (3.11).

Let the scaled nominal values of parameters be $C_1 = 20$, $r_x = 10$, $r_\pi = 40$, $C_u = 25$, $C_2 = 20$, $R_s = 75$, $R_e = 30$, $C_\pi = 15$, $C_e = 10$, $g_m = 10$, $R_c = 10$, $R_l = 20$. We then simulated the system on the CDC - 1604 computer, using twelve new sets of parameters with one component value changed at a time. The algorithm used is the Golden section search method and the function to be minimized is

Table 1. Computer Simulation Results for the Butterworth Filter

	C_1	L_1	C_2	L_2	C_1	L_1	C_2	L_2
New Parameter Values	20	20	30	40	10	40	30	40
t'	20	30.66	39.87	51.89	28.75	40	48.5	62.2
$f(t')$	1.25×10^{-12}	5.88×10^{-8}	8.46×10^{-9}	1.96×10^{-6}	1.62×10^{-7}	2.10×10^{-14}	2.42×10^{-7}	6.94×10^{-6}
New Parameter Values	10	20	50	40	10	20	30	45
t'	30.27	41.62	50	64.01	14.18	24.46	34.13	45
$f(t')$	2.91×10^{-8}	2.73×10^{-7}	1.47×10^{-13}	6.87×10^{-6}	3.93×10^{-7}	4.93×10^{-7}	4.13×10^{-7}	1.04×10^{-13}

$$\begin{array}{c}
 V_{c1} \\
 V_{rx} \\
 V_{r\pi} \\
 V_{cu} \\
 V_{c2} \\
 I_{RS} \\
 I_{Re} \\
 I_{c\pi} \\
 I_{Ce} \\
 I_{gm} \\
 I_{Rc} \\
 I_{RL}
 \end{array}
 =
 \begin{array}{cccccccccccc}
 1/c_{1s} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & r_x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & r_{\pi} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1/c_{us} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 1/c_{2s} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1/R_s & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1/R_e & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_{\pi s} & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_{es} & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & g_m & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/R_c & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/R_L
 \end{array}
 \begin{array}{c}
 I_{c1} \\
 I_{rx} \\
 I_{r\pi} \\
 I_{cu} \\
 I_{c2} \\
 V_{RS} \\
 V_{Re} \\
 V_{c\pi} \\
 V_{Ce} \\
 V_{gm} \\
 V_{Rc} \\
 V_{RL}
 \end{array}
 \quad (3.10)$$

$$\text{and } \begin{array}{c} I_{c1} \\ I_{rx} \\ I_{r\pi} \\ I_{cu} \\ I_{c2} \\ V_{RS} \\ V_{Rc} \\ V_{c\pi} \\ V_{ce} \\ V_{gm} \\ V_{Rc} \\ V_{RL} \\ V_o \end{array}
 =
 \begin{array}{cccccccccccc|c}
 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
 -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
 -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 -1 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
 -1 & -1 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
 -1 & -1 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
 \end{array}
 \begin{array}{c}
 V_{c1} \\
 V_{rx} \\
 V_{r\pi} \\
 V_{cu} \\
 V_{c2} \\
 I_{RS} \\
 I_{Re} \\
 I_{c\pi} \\
 I_{Ce} \\
 I_{gm} \\
 I_{Rc} \\
 I_{RL} \\
 V_i
 \end{array}
 \quad (3.11)$$

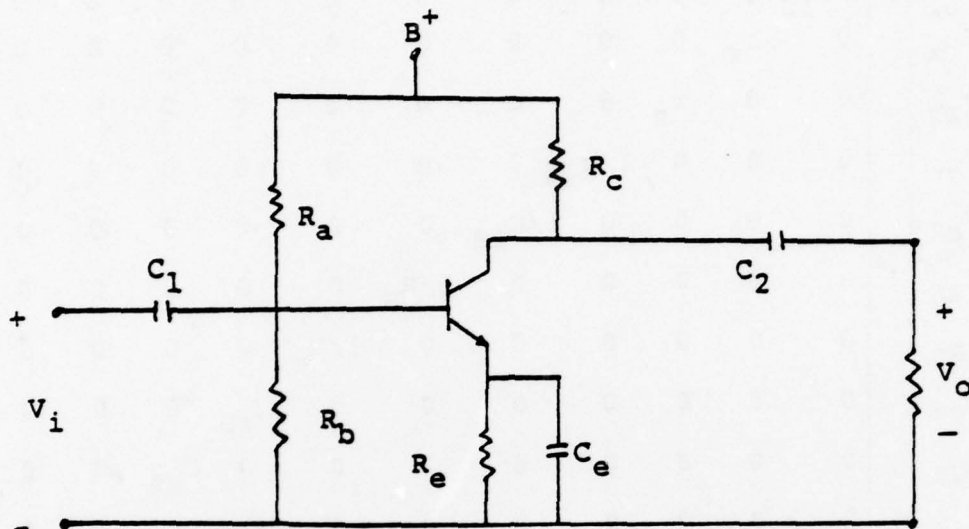


Figure 3. One stage transistor amplifier.

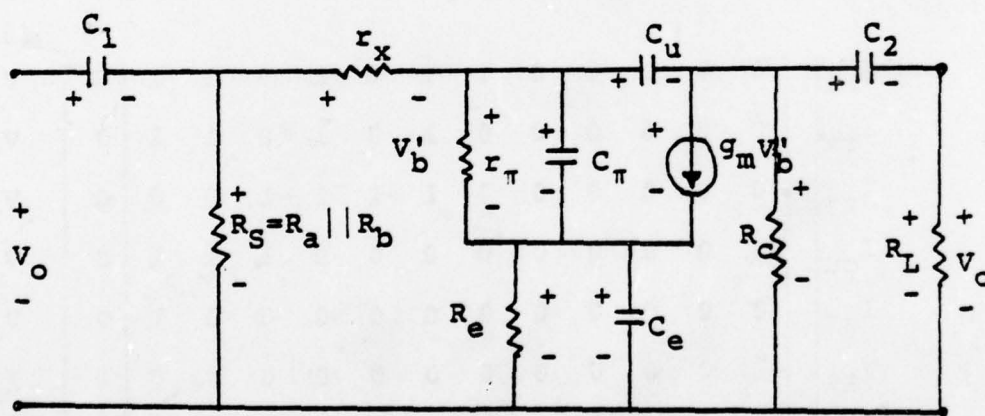


Figure 4. Amplifier equivalent circuit

$$f(t) = ||S_1(s,t) - S(s)||^2$$

where t denotes the new value of the faulty component.

The results are summarized in table 2. From table 2, the conclusion is again that for the failed component, $f(t')$ approaches zero, and t' is exactly equal to component value. Note that if the minimum occurs at infinity, we assume it is at the point 1000 in the computer simulation process.

Table 2. Computer Simulation Results for one Stage Transistor Amplifier.

C_1	r_x	r_{π}	C_u	C_2	R_S	R_e	C_{π}	C_e	g_m	R_C	R_L
New Actual value 50	10	40	25	20	75	30	15	10	10	10	20
t' 50	88.5	0.44	28.77	1000	1000	62.6	80.1	9.2	4.5	18.2	41.9
$f(t')$ 6.2×10^{-4}	2.6×10^{-4}	3.1×10^{-5}	3.0×10^{-5}	1.58×10^{-5}	4.5×10^{-5}	5.3×10^{-5}	3.0×10^{-5}	1.7×10^{-5}	3.4×10^{-5}	1.2×10^{-5}	1.8×10^{-5}
New Actual value 20	30	40	25	20	75	30	15	10	10	10	20
t' 1.68	30	1000	14.6	0.9	0.96	7.56	0.1	16.9	1000	1.2	2.0
$f(t')$ 1.1×10^{-3}	3.2×10^{-2}	2.1×10^{-2}	2.1×10^{-3}	1.3×10^{-3}	1.3×10^{-3}	8.5×10^{-3}	1.9×10^{-3}	3.5×10^{-3}	1.6×10^{-3}	7.4×10^{-3}	9.0×10^{-3}
New Actual value 20	10	90	25	20	75	30	15	10	10	10	20
t' 19.9	10.01	90	24.97	19.8	72.7	29.9	14.6	10.01	10.09	9.96	19.9
$f(t')$ 1.1×10^{-3}	7.8×10^{-3}	7.6×10^{-20}	3.3×10^{-11}	2.1×10^{-8}	2.2×10^{-8}	5.2×10^{-8}	1.2×10^{-8}	4.1×10^{-8}	1.3×10^{-8}	6.2×10^{-8}	6.9×10^{-8}
New Actual value 20	10	40	60	20	75	30	15	10	10	10	20
t' 1000	5.79	0.1	60	1000	1000	1000	1000	6.65	0.72	1000	1000
$f(t')$ 5.6×10^{-3}	3.4×10^{-3}	3.0×10^{-3}	8.1×10^{-13}	9.5×10^{-3}	1.4×10^{-3}	6.0×10^{-3}	1.7×10^{-3}	2.0×10^{-3}	3.1×10^{-3}	2.7×10^{-3}	4.2×10^{-3}
New Actual value 20	10	40	25	50	75	30	15	10	10	10	20
t' 30.9	9.4	0.88	26.9	50	1000	43	43	9.6	6.3	13.5	28.7
$f(t')$ 4×10^{-9}	9.7×10^{-5}	1.6×10^{-5}	1.7×10^{-5}	1.7×10^{-4}	8×10^{-5}	6.3×10^{-5}	1.2×10^{-4}	6.8×10^{-5}	1.2×10^{-4}	1.4×10^{-4}	6.9×10^{-4}

Cont.	C_1	r_x	r_π	C_u	C_2	R_s	R_e	C_π	C_e	g_m	R_c	R_L
New Actual value	20	10	40	25	20	15	30	15	10	10	10	20
t'	12.3	11.1	1000	22.6	9.5	15	20.5	0.1	10.8	50.9	6.9	13
$f(t')$	5.2×10^5	1.6×10^4	4.7×10^3	2.1×10^5	4.1×10^8	9.1×10^3	3.2×10^5	3.1×10^4	1×10^4	2.4×10^4	1.8×10^5	9.4×10^8
New Actual value	20	10	40	25	20	75	80	15	10	10	10	20
t'	61.4	8.9	0.4	29.2	1000	1000	80	78.3	9.2	4.6	19.6	47.8
$f(t')$	6.8×10^5	5.5×10^4	1.5×10^4	1.5×10^4	4.3×10^5	6.6×10^4	2.2×10^4	5.9×10^4	4.2×10^4	6.4×10^4	1.4×10^4	8.8×10^5
New Actual value	20	10	40	25	20	75	30	55	10	10	10	20
t'	27.4	9.2	1	26.7	34.6	1000	37.1	55	9.5	5.3	12.8	26.4
$f(t')$	8.5×10^5	2.5×10^6	5.4×10^5	5.2×10^5	1.1×10^4	1.3×10^4	1.3×10^4	3.5×10^4	1×10^5	9.6×10^7	6.3×10^5	7.6×10^5
New Actual value	20	10	40	25	20	75	30	15	30	10	10	20
t'	0.1	1000	1000	10	0.1	0.1	2.7	0.1	30	1000	0.1	0.1
$f(t')$	2.6×10^3	2.4×10^3	7.6×10^2	6.1×10^4	3.1×10^3	3.1×10^3	1.3×10^2	7.3×10^2	1.9×10^{11}	6.7×10^2	2.9×10^3	2.5×10^3
New Actual value	20	10	40	25	20	75	30	15	10	40	10	20
t'	15.4	10.9	1000	23.6	13.5	27.6	25.4	0.1	10.5	40	8.1	15.8
$f(t')$	8.9×10^5	9.4×10^6	2×10^4	6.2×10^5	1.1×10^4	1.1×10^4	1.3×10^4	5.7×10^5	2×10^5	5.7×10^5	7×10^5	8×10^5

Cont.	C_1	r_x	r_π	C_u	C_2	R_s	R_e	C_π	C_e	g_m	R_C	R_L
New Actual												
value	20	10	40	25	20	75	30	15	10	10	50	20
t'	1000	7.8	0.2	33.9	1000	1000	1000	1000	8.4	2.4	50	1000
$f(t')$	6.9×10^5	8.3×10^4	6.7×10^5	5.3×10^5	8.4×10^4	2.3×10^3	3.4×10^4	2.3×10^3	5.1×10^4	1.1×10^4	4.3×10^4	6.2×10^4
New Actual												
value	20	10	40	25	20	75	30	15	10	10	10	60
t'	84	8.5	0.3	30.3	1000	1000	93.9	1000	8.9	3.6	23.7	60
$f(t')$	3.3×10^5	4.2×10^4	4.1×10^5	3.7×10^5	1.3×10^4	9.1×10^4	1.1×10^4	3.9×10^3	2.7×10^4	5.6×10^4	9.1×10^4	4.7×10^4

CHAPTER IV
JACOBIAN TECHNIQUE SOLVING FOR
THE FAULT DIAGNOSIS EQUATIONS

Using the matrix identity,

$$\frac{\partial}{\partial \mathbf{x}} \mathbf{Y}^{-1} = -\mathbf{Y}^{-1} \frac{\partial \mathbf{Y}}{\partial \mathbf{x}} \mathbf{Y}^{-1}, \quad (4.1)$$

it is easily shown that the derivative of the connection function, $S(s, \bar{r})$, can be expressed as

$$\frac{\partial S(s, \bar{r})}{\partial r_i} = L_{21} (I - Z(s, \bar{r}) L_{11})^{-1} \frac{\partial Z(s, \bar{r})}{\partial r_i} [I + L_{11} (I - Z(s, \bar{r}) L_{11})^{-1} Z(s, \bar{r})] L_{12} \quad (4.2)$$

Equation (4.2) implies that the Jacobian matrix of the fault diagnosis equations can be computed analytically with no additional matrix inversion involved over and above that required to evaluate the connection function suggesting that we might solve the fault diagnosis equations by way of the Jacobian matrix. To solve the fault diagnosis equations under the assumptions that all parameter variations are small and that the dynamical properties of the components remain unchanged, it is expected that a single iteration of the Newton-Raphson method will yield a good approximation to the solution of the equations in the form

$$\bar{r}_i \approx \bar{r}_0 - J_{\underline{s}}^{-1}(\bar{r}_0) (\underline{s}(\bar{r}_0) - \underline{s}^m) \quad (4.3)$$

Here, \bar{r}_0 is the nominal parameter vector,

$$\underline{s}(\bar{r}_0) = \text{col}(\text{vec} S_i(\bar{r}_0)), \quad \underline{s}^m = \text{col}(\text{vec} S_i^m),$$

and S_i^m is the measured frequency response at the complex frequency S_i , and $J_{\underline{s}}(\bar{r}_0)$ is the Jacobian matrix of \underline{s} evaluated at \bar{r}_0 . The "vec" operation used above is formalized by the equality

$$\text{vec}(A) = \text{col}(a_i) \quad (4.4)$$

where a_i denotes the i^{th} column of the matrix A .

From equation (4.3), we have

$$\bar{e} = \bar{r}_0 - \bar{r}_1 \approx J_{\underline{s}}^{-1}(\bar{r}_0) (\underline{s}(\bar{r}_0) - \underline{s}^m) \quad (4.5)$$

where \bar{e} is the error vector between the nominal parameter vector and the new parameter vector. If the solution is close enough to \bar{r}_0 , it is expected that the entries of the \bar{e} vector will consist of two disjoint subsets separated by a threshold, T ; the elements in one set, E_1 , are always greater than T , while the elements of the other set, E_2 , are always less than T . It follows from a heuristic argument that the failed components are subset of E_1 . Thus, the dimension of the original problem has been reduced

to that of E_1 , and the failure can be diagnosed easily because of the small dimensionality of E_1 . The advantage of such an approach is that all computations of $\underline{S}(\bar{F}_0)$ and $J_{\underline{S}}^{-1}(\bar{F}_0)$ are at the nominal parameter vector \bar{F}_0 , which can be precomputed. This implies that the algorithm may be implemented on a real time basis for a large interconnected dynamical system.

The above algorithm has been simulated for several systems on the CDC-1604 computer, and the results seem a little restricted. As a first example, consider the Butterworth filter shown in figure 2, for which the component connection model takes the form of equations (3.8) and (3.9). The nominal values of parameters are $C_1 = 10$, $L_1 = 20$, $C_2 = 30$, $L_2 = 40$, and the computer simulation results of the example are summarized in Table 3. From table 3, it is observed that for small changes in the parameters, this algorithm yields good results.

As a more sophisticated example, the dimension of the Butterworth filter is increased to eight components. The component connection model then becomes as shown in (4.6) and (4.7).

Simulating the system on the computer using the nominal values $C_1 = 10$, $L_1 = 20$, $C_2 = 30$, $L_2 = 40$, $C_3 = 35$, $L_3 = 25$, $C_4 = 15$, $L_4 = 5$, the results are summarized in Table 4.

Table 3. Computer Simulation Results for the 4th order Butterworth Filter.

	C_1	L_1	C_2	L_2	C_1	L_1	C_2	L_2
New Parameter Values	11	20	30	40	10	21	30	40
Error Vector Values	-9.6×10^{-3}	2.6×10^{-2}	-1×10^{-3}	5.5×10^{-2}	4.9×10^{-2}	-9.7×10^{-1}	-2.4×10^{-2}	-5.1×10^{-2}
New Parameter Values	10	20	31	40	10	20	30	41
Error Vector Values	3.4×10^{-3}	3.2×10^{-2}	-9.7×10^{-1}	-5.6×10^{-2}	1.9×10^{-2}	4.9×10^{-2}	2.3×10^{-2}	-1.03

AD-A060 941

TEXAS TECH UNIV LUBBOCK INST FOR ELECTRONICS SCIENCE
FAULT ANALYSIS IN ELECTRONIC CIRCUITS AND SYSTEMS. II.(U)
JAN 78 R SAEKS, N SEN, H M CHEN, K S LU

F/6 9/3

N00014-75-C-0924

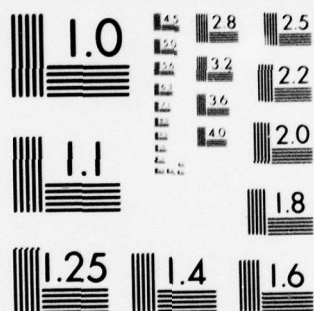
UNCLASSIFIED

NL

4 OF 4

AD
A080841





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

$$\begin{bmatrix} v_{C1} \\ i_{L1} \\ v_{C2} \\ i_{L2} \\ v_{C3} \\ i_{L3} \\ v_{C4} \\ i_{L4} \end{bmatrix} = \begin{bmatrix} 1/s_{C1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/s_{L1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/s_{C2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/s_{L2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/s_{C3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/s_{L3} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/s_{C4} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/s_{L4} \end{bmatrix} \begin{bmatrix} i_{C1} \\ v_{L1} \\ i_{C2} \\ v_{L2} \\ i_{C3} \\ v_{L3} \\ i_{C4} \\ v_{L4} \end{bmatrix} \quad (4.6)$$

and

$$\begin{bmatrix} i_{C1} \\ v_{L1} \\ i_{C2} \\ v_{L2} \\ i_{C3} \\ v_{L3} \\ i_{C4} \\ v_{L4} \\ \hline v_o \end{bmatrix} = \begin{bmatrix} -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & | & 1 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & | & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & | & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & | & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & | & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & | & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & | & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & | & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & | & 0 \end{bmatrix} \begin{bmatrix} v_{C1} \\ i_{L1} \\ v_{C2} \\ i_{L2} \\ v_{C3} \\ i_{L3} \\ v_{C4} \\ i_{L4} \\ \hline v_i \end{bmatrix} \quad (4.7)$$

Table 4. Computer Simulation Results for the 8th order Butterworth Filter

	C_1	L_1	C_2	L_2	C_3	L_3	C_4	L_4
New Parameter Values	10.1	20	30	40	35	25	15	5
Error Vector Values	0.5	8.01	5.45	7.6	-9.43	-10.23	-1.62	-0.26

As shown in the table, even for a very small amount of change, the error vector does not manifest the phenomenon which we expect. Thus this algorithm is not suitable for general purpose applications.

The reason that the method fails is that the Jacobian matrix for the fault diagnosis equations is usually ill-behaved for matrix inversion. The larger the dimension of the system, the more numerical errors occur; thus the technique works well for the small system but fails in the larger system. A technique for improving the shortcomings of this method is based on the fact that the structure of the Jacobian matrix for the fault diagnosis equations strongly depends upon the frequencies we select to perform the multifrequency test. Some experimental work on the computer has shown that if we select frequencies such that the Jacobian matrix becomes row dominant, the results will be more satisfactory. The mechanism behind this technique

deserves further investigations.

CHAPTER V

CONCLUSION

In this thesis, several search algorithms for the solutions of the fault diagnosis equations have been developed. Although the applications of the algorithms are restricted to the case of one error or one group of errors, it can be extended to the multivariable case with some suitable modifications. This approach has two important characteristics, first, by using Householder formula, only a very small matrix is inverted in each iteration, thus significantly reducing computer time; second, the success of this algorithm brings us closer to the goal of automated fault diagnosis.

In recent years many papers concerning the fault diagnosis work have been presented, but to our knowledge the search algorithm is the best thus far obtained for the solution of the diagnosis equations with a single fault. Yet, much work still has to be done in this area. The ill-conditioned behavior of the Jacobian matrix of the fault diagnosis equations makes it very hard to solve the equations by standard algorithm such as Newton's method. However, the freedom in the selection of test frequencies makes it possible to change the structure of the Jacobian matrix and thus open up the possibility of improving the

Jacobian technique for the solutions of the fault diagnosis equations.

REFERENCES

1. J. C. Rault, "A Bibliography on Fault Detection and Location in Digital Systems," Rational Fault Analysis, Marcel Dekker, New York, 1977.
2. J. C. Rault, R. Garzia and S. D. Bedrosian, "Fault Detection and Location in Analog Circuits - A Bibliography," Rational Fault Analysis, Marcel Dekker, New York, 1977.
3. M. N. Ransom and R. Saeks, "The Connection Function Theory and Application," Inter. Journal of Circuit Theory and Applications, Vol. 3, pp 5-21, 1975.
4. H. Neudecker, "Some Theorems on Matrix Differentiation with Special Reference to Kronecker Matrix Products," Am. Stat. Assoc. J., pp. 953-963, Sept. 1969.
5. R. P. Tewarson, "Sparse Matrices," Academic Press, New York, 1973.
6. S. MacLane, and G. Birkhoff, "Algebra," Macmillan, New York, 1967.
7. E. Polak, "Computational Methods in Optimization," Academic Press, New York, 1971.
8. P. E. Gray and C. I. Searle, "Electronic Principles-Physical Models and Circuits," Wiley, New York, 1969.
9. E. Isaacson and H. B. Keller, "Analysis of Numerical Methods," Wiley, New York, 1966.
10. R. Saeks, S. P. Singh and R. W. Liu, "Fault Isolation via Components Simulation," IEEE Trans. on Circuit Theory, Vol. CT-19, pp 634-640, 1972.

ECMP - A SOFTWARE PACKAGE FOR COMPUTATION OF MEASURABILITY
AND COST TO AID TEST POINT SELECTION*

N. Sen**

*This research supported in part by Office of Naval Research Contracts
75-C-0924 and 76-C-1136.

**Presently with Datapoint Corporation, San Antonio, Texas.

I. Introduction

Package ECMP* is a FORTRAN 63 coded routine that will compute the value of the DELTA (δ) parameter, which represents a measure of testability for electronic circuits and systems. It will also output the cost involved corresponding to each possible combination of inputs and outputs specified.

The inputs to the package consist of the Component-Connection Model of the circuit or system and certain other information corresponding to costs, frequencies, formats and options. The standard output consist of each case and the corresponding cost, δ , and the last test eigenvalue. Several options (both on input and output) are available to the user for various changes.

For a description of the analytical aspects behind the package, user is referred to references [1] - [3]. These, especially, Chapter IV in reference [3], will help to give a better understanding to the user.

The package needs no modification from problem to problem and is only limited by the memory size or computer time. A short calling program, two subroutines and a data deck to require modification by the user from problem to problem, as they supply both the analytical representation of a particular problem and the information dictating operating modes, initial conditions for the general subroutine, ECMP and the options in effect.

ECMP is written for single precision calculations to work on a CDC 1604 computer. Suitable modifications must be made to comply with other computer systems, if to be used. Also, changes are required for use of double pre-

* Electronic Circuit Measurability Package

cision. This may mean modification to the general routine ECMP and to the subroutines contained in the ECMP.

An explanation of the user supplied routines and data will now be given. This will be followed by a complete program listing in section 3.

Finally, section 4 contains an example run on the package which includes sample user supplied subroutines and the standard output printout.

II. User Supplied Routines

A brief description and form listing of each subroutine is given in this section.

2.1 Main Calling Program

The user supplied main program properly dimensions all matrices used by the general routine ECMP, and makes the appropriate call to ECMP, thus passing control to the general routine. The calling program is:

```

DIMENSION  A(NZ,NZ), AL11(NZ,NZ), AL12(NZ,NZ), AL21(NO,NZ),
AL22(NO,NI), B(NZ,NZ), C(NZ,NZ), CCOF(NFVEC,NP),
COF(NF,NVEC,NP), COL(NVEC),COST1(NI), COSTO(NO), D(NZ,NZ),DIFZ(NZ,NZ),
DP(NP,NP), E(NZ,NZ), F(NZ,NZ), G(NZ,NZ),
H(NZ,NZ), LCOMI (MAXCAS), LCOMO(MAXCAS), NUMB(MAXITER),
ODG(NP), P(NO,NZ), R(NO,NZ), SDG(NP), SENS(NF,NVEC,NP),
SEQ(NSEQ), SYSM(NO,NI), T(NO,NI), TCCOF(NP,NFVEC),
VAL(MAXITER), VRF(10), VWF(10), W(NZ,NI), XI(NZ,NZ), Z(NZ,NZ)

READ (5,1000) INDOPT, MAXCAS, MAXITER, NF, NFVEC, NI, NO, NP,
           NSEQ, NVEC, NZ.

CALL  ECMP (A, AL11, AL12, AL21, AL22, B, C, CCOF, COF, COL, COST1,
           COSTO, D, DIFZ, DP, E, F, G, H, INDOPT,
           LCOMI, LCOMO, MAXCAS, MAXITER, NF, NFVEC, NI, NO, NP,
           NSEQ, NUMB, NVEC, NZ, ODG, P, R, SDG, SENS, SEQ, SYSM,
           T, TCCOF, VAL, VRF, VWF, W, XI, Z)

1000 FORMAT (11 I 5)

END.
```

The arguments in the dimension statements are only symbolic; specific integer values equal to these values must be used in the dimension statements

when executing a program. Also integer values are to be near for Data Card 1 in integer FORMAT.

An explanation of the 48 symbols follows:

A = WORKING MATRIX OF SIZE $NZ \times NZ$
 AL11 = FEEDBACK MATRIX L11 OF SIZE $NZ \times NZ$
 AL12 = POST COMPENSATOR MATRIX L12 OF SIZE $NZ \times NI$
 AL21 = PRE COMPENSATOR MATRIX L21 OF SIZE $NO \times NZ$
 AL22 = FEED FORWARD MATRIX L22 OF SIZE $NO \times NI$
 B = WORKING MATRIX OF SIZE $NZ \times NZ$
 C = WORKING MATRIX OF SIZE $NZ \times NZ$
 CCOF = COEFFICIENT MATRIX OF SIZE $NFVEC \times NP$
 COF = COEFFICIENT ARRAY OF SIZE $NF \times NVEC \times NP$
 COL = COLUMN VECTOR OF SIZE $NVEC$
 COSTI = VECTOR OF SIZE NI TO CONTAIN INPUT COSTS
 COSTO = VECTOR OF SIZE NO TO CONTAIN OUTPUT COSTS
 D = WORKING MATRIX OF SIZE $NZ \times NZ$
 DIFZ = SINGLE PARAMETER SENSITIVE COMPONENT MATRIX OF SIZE $NZ \times NZ$
 DP = SYMMETRIC POSITIVE SEMIDEFINITE MATRIX OF SIZE $NP \times NP$
 E = WORKING MATRIX OF SIZE $NZ \times NZ$
 F = WORKING MATRIX OF SIZE $NZ \times NZ$
 G = WORKING MATRIX OF SIZE $NZ \times NZ$
 H = WORKING MATRIX OF SIZE $NZ \times NZ$
 INDOPT = INDICATOR FOR ANY OPTIONAL FEATURE USED OR DESIRED. SHOULD BE: ANY NON-ZERO INTEGER, IF ANY OPTIONAL FEATURE IS DESIRED (CORRESPONDING TO INDX1 TO INDX12). EXPLANATION OF THESE FEATURES IS ON THE NEXT DATA CARD DETAILS
 LCOMI = VECTOR OF SIZE $MAXCAS$ TO CONTAIN INPUT COMBINATIONS
 LCOMO = VECTOR OF SIZE $MAXCAS$ TO CONTAIN OUTPUT COMBINATIONS
 MAXCAS = MAXIMUM NUMBER OF CASES TO BE ANALYZED = $(2^{**NO}-1) * (2^{**NI}-1)$
 MAXITER = MAXIMUM NUMBER OF ITERATIONS TO BE PERFORMED AT AN INTERNAL OF 10^{-1} STARTING FROM AN ASSUMED TEST EIGENVALUE OF 1.0 FOR STURM SEQUENCE COMPUTATION
 NF = NUMBER OF FREQUENCIES TO BE USED FOR ANALYSIS
 NFVEC = NUMBER OF ELEMENTS IN THE VECTOR OF INPUTS AND OUTPUTS AT ALL FREQUENCIES = $NO \times NI \times NF$
 NI = NUMBER OF INPUTS
 NO = NUMBER OF OUTPUTS
 NP = NUMBER OF POTENTIALLY VARIABLE CIRCUIT PARAMETERS ALSO CALLED SENSITIVITY PARAMETERS
 NSEQ = NUMBER OF ELEMENTS IN THE STURM SEQUENCE VECTOR = $NP+1$
 NUMB = MAXITER TO CONTAIN NUMBER OF EIGENVALUES LESS THAN TEST VALUE FOR EACH ITERATION
 NVEC = NUMBER OF ELEMENTS IN THE VECTOR OF INPUTS AND OUTPUTS AT EACH FREQUENCY = $NO \times NI$
 NZ = NUMBER OF COMPONENTS
 ODG = VECTOR OF SIZE NP TO CONTAIN ON DIAGONAL ELEMENTS OF THE TRIDIAGONAL MATRIX
 P = WORKING MATRIX OF SIZE $NO \times NZ$
 R = WORKING MATRIX OF SIZE $NO \times NI$

SDG = VECTOR OF SIZE NP TO CONTAIN SUPER OR SUB
 DIAGONAL ELEMENTS OF THE TRIDIAGONAL MATRIX
 SENS = ARRAY OF SIZE NF*NVEC*NP TO CONTAIN
 JACOBIAN MATRIX
 SEQ = VECTOR OF SIZE NSEQ TO CONTAIN STURM SEQUENCE
 SOLUTION FOR A PARTICULAR ITERATION
 SYSM = COMPOSITE SYSTEM TRANSFER FUNCTION
 MATRIX OF SIZE NO*NI
 T = WORKING MATRIX OF SIZE NO*NI
 TCCOF = MATRIX OF SIZE NP*NVEC TO CONTAIN
 TRANSPOSED COEFFICIENTS
 VAL = VECTOR OF SIZE MAXITER TO CONTAIN TEST
 EIGENVALUE
 VRF = VARIABLE OF SIZE 10 TO CONTAIN VARIABLE
 READ FORMAT
 VWF = VARIABLE OF SIZE 10 TO CONTAIN VARIABLE
 WRITE FORMAT
 W = WORKING MATRIX OF SIZE NZ*NI
 XI = IDENTITY MATRIX OF SIZE NZ*NZ
 Z = COMPONENT MATRIX OF SIZE NZ*NZ

Caution must be exercised in including all the 37 matrices in the DIMENSION statement and in all the 48 entries in the CALL statement argument list properly. In the remaining user supplied subroutines, all matrices are given the variable dimensions as allowed by FORTRAN.

2.2 Subroutine For Supplying Composite Component Model

(Subroutine COMPNT)

This user supplied subroutine COMPNT lists all the components to be used in the Z matrix element by element. Any element from the total size of the matrix NZ X NZ which has not been specified in this routine, is automatically assumed to be zero in the package (by subroutine ECMP).

An example of this subroutine for a 12 component case follows: Here it is assumed that the composite component model has been almost reduced to a diagonal form. It has been shown [4], that this is always possible though this is not a limitation imposed by the present package. The package only expects all nonzero entries to be listed whether diagonal or off-diagonal.

The right hand side (RHS) of each of the equality entry below should

contain the actual numerical value of the component (may be nominal) or a variable s (real frequency). Frequency s is the only variable permitted on the RHS unless the value of any other variable, if used, has been explicitly defined in this subroutine.

No changes should be made in the following statements:

```
SUBROUTINE COMPNT (NZ, s, Z)
```

```
  DIMENSION  Z(NZ, NZ)
```

Example Subroutine COMPNT (deck)

```
SUBROUTINE COMPNT (NZ, s, Z)
```

```
  DIMENSION Z(NZ,NZ)
```

```
  Z (1,1) = 1.0/s
```

```
  Z (2,2) = 1.0
```

```
  Z (3,3) = 1.0
```

```
  Z (4,4) = 1.0/s
```

```
  Z (5,5) = 1.0/s
```

```
  Z (6,6) = 1.0
```

```
  Z (7,7) = 1.0
```

```
  Z (8,8) = s
```

```
  Z (9,9) = s
```

```
  Z (10,8) = 1.0
```

```
  Z (11,11) = 1.0
```

```
  Z (12,12) = 1.0
```

```
  RETURN
```

```
  END
```

2.3 Subroutine for Supplying Derivative of the Component Matrix Z with Respect to the Potentially Variable Circuit Parameter

(Subroutine SENSTVTY)

This subroutine will supply the DIFZ Matrix to the package ECMP and needs special attention. Since ECMP is supposed to compute Jacobian corresponding to each of the potentially variable circuit parameter separately, this subroutine will have NP calls from the ECMP where NP is the total number of potentially variable circuit parameters. Hence, for each call, the user has to supply the information in the following format where each call must be able to supply all the nonzero entries in the DIFZ matrix corresponding to only one of the potentially variable circuit parameters. Again, all the entries not supplied will be assumed zero in each case by ECMP. Also, just like the SUBROUTINE COMPNT, no entry in the subroutine given below, on the right hand side (RHS) of equality may contain any other variable except s unless defined within this subroutine.

Example Subroutine SENSTVTY (deck) for 12 Parameter case (NP = 12) where for each sensitivity parameter only one entry gets changed.

```

SUBROUTINE SENSTVTY (DIFZ,KS,NZ,S)
  DIMENSION DIFZ (NZ,NZ)
  GO TO (1,2,3,4,5,6,7,8,9,10,11,12), KS
1  DIFZ (1,1) = 1.0/s
  GO TO 100
2  DIFZ (2,2) = 1.0
  GO TO 100
3  DIFZ (3,3) = 1.0
  GO TO 100
4  DIFZ (4,4) = -1.0/s
  GO TO 100

```



```
5  DIFZ (5,5) = -1.0/s
   GO TO 100
6  DIFZ (6,6) = -1.0
   GO TO 100
7  DIFZ (7,7) = -1.0
   GO TO 100
8  DIFZ (8,8) = s
   GO TO 100
9  DIFZ (9,9) = s
   GO TO 100
10 DIFZ (10,8) = 1.0
   GO TO 100
11 DIFZ (11,11) = -1.0
   GO TO 100
12 DIFZ (12,12) = 1.0
100 RETURN
   END.
```

It is quite evident from the above example, that for a circuit with NP sensitivity parameters the computed GO TO statement will be of the form :

GO TO (1,2,3,4, NP), KS

and there will be NP statement numbers in addition to the statement number 100 (RETURN) from 1 to NP giving information on non zero entries, corresponding to NP successive calls.

2.4 Data Card Set

2.4.1 Data Card 1.

Corresponding to

```
READ (5,1000) INDOPT, MAXCAS, MAXITER, NF, NFVEC, NI,  
              NO, NP, NSEQ, NVEC, NZ
```

```
1000  FORMAT (11I5)
```

in Main Program (User Supplied)

This card shall contain integer values corresponding to the above 11 variables. An explanation of these variables has been given earlier in Section 2.1.

2.4.2 Data Card 2. (To be supplied, if INDOPT on Data Card 1 is any non zero integer)

Corresponding to

```
READ (5,1010) INDX1, INDX2, INDX3, INDX4, INDX5, INDX6,  
              INDX7, INDX8, INDX9, INDX10, INDX11, INDX12
```

```
1010  FORMAT (12I5)
```

in Subroutine ECMP.

This card (if supplied - depending on INDOPT) shall contain 12 integers any of which if non zero will indicate the specific option exercised by the user. These symbols INDX 1 to INDX 12 correspond to the following 12 options respectively (for non zero integer entries)

INDX 1 : - Variable Format for Inputting connection matrices (AL11, AL12, AL21, AL22)
and Input and Output costs vectors (COST1, COST0)
desired.

INDX 2 : - Variable Format for Outputting connection matrices (AL11, AL12, AL21, AL22),
Z Matrix,
DIFZ Matrix,
SYSM Matrix for all frequencies,
SENS Matrix,
COF Matrix for each case,
DP Matrix for each case desired.

INDX 3 : - Option on starting frequency (SFREQ) and step size (STEP) exercised.

INDX 4 : - Printout of Input data desired NZ, NI, NO, NP, NF, NVEC, NFVEC,
NSEQ, MAXCAS, MAXITER, SFREQ, STEP,
Connection Matrices (AL11, AL12, AL21, AL22)
Input and Output cost vectors (COST1, COST0)

INDX 5 : - Printout on Z Matrix at all frequencies (will be NF matrices).

INDX 6 : - Printout on Computer System Transfer for SYSM for all frequencies desired
(will be NF Matrices).

INDX 7 : - Printout on DIFZ Matrix for all parameters (will be NF X NP matrices) and
for all frequencies desired.

INDX 8 : - Printout of Jacobian SENS desired.

INDX 9 : - Printout of COF Matrix for each case desired.

INDX 10 : - Printout of DP Matrix for each case desired.

INDX 11 : - Printout of ODG, SDG and Sturm Sequence solution for each-assumed test
eigenvalue desired.

INDX 12 : - Printout of count of eigenvalues for each test value in Sturm sequence
solution desired.

2.4.3 Data Card 3 (To be supplied if INDX 1 on Data Card 2 is non zero integer)

Corresponding to

READ (5,1020), (VRF (I), I = 1,10)

1020 FORMAT (10A8)

in Subroutine ECMP

Reads Variable Read Format (VRF) if option 1 is exercised.

Default value : - (12F6.0)

2.4.4 Data Card 4 (To be supplied if INDX 2 on Data Card 2 is non zero integer)

Corresponding to

READ (5, 1020) (VWF (I), I = 1,10)

1020 FORMAT (10A8)

in Subroutine ECMP

Reads Variable Write Format (VWF) if option 2 is exercised.

Default value : - (6E20.3)

2.4.5 Data Card 5 (To be supplied if INDX 3 on Data Card 1 is non zero integer)

Corresponding to

READ (5,1050) SFREQ, STEP

1050 FORMAT (2F6.0)

in Subroutine ECMP

Reads Starting Frequency (SFREQ), and Step Size (STEP) if option 3 is exercised.

Default values: - SFREQ = 1.0

STEP = 1.0

2.4.6 Data Card(s) 6 (Must be supplied)

Corresponding to

READ (5, VRF) ((AL11 (I,J), J = 1, NZ), I = 1, NZ)

or

READ (5,1080) ((AL11 (I,J), J = 1, NZ), I = 1, NZ)

1080 FORMAT (12F6.0)

in Subroutine ECMP

Reads AL11 matrix depending on the variable format if option 1 is exercised.

Otherwise in Format (12F6.0) in case of Default.

2.4.7 Data Card(s) 7 (Must be supplied)

Corresponding to

READ (5, VRF) ((AL12, (I,J), J = 1, NI), I = 1, NZ)

or

READ (5,1080) ((AL12 (I,J), J = 1, NI), I = 1, NZ)

1080 FORMAT (12F6.0)

in Subroutine ECMP.

Reads AL12 matrix depending on the variable format if option 1 is exercised.

Otherwise in Format (12F6.0) in case of Default.

2.4.8 Data Card(s) 8 (Must be supplied)

Corresponding to

READ (5,VRF) ((AL21 (I,J), J = 1, NZ), I = 1, NO)

or

READ (5,1080) ((AL21 (I,J), J = 1, NZ), I = 1, NO)

1080 FORMAT (12F6.0)

in Subroutine ECMP.

Reads AL21 matrix depending on the variable format if option 1 is exercised.

Otherwise in Format (12F6.0) in case of Default.

2.4.9 Data Card(s) 9 (Must be supplied)

Corresponding to

READ (5, VRF) ((AL22 (I,J), J = 1, NI), I = 1, NO)

or

READ (5, 1080) ((AL22 (I,J), J = 1, NI), I = 1, NO)

1080 FORMAT (12F6.0)

in subroutine ECMP.

Reads AL22 matrix depending on the variable format if option 1 is exercised.

Otherwise in format (12F6.0) in case of Default.

2.4.10 Data Card(s) 10 (Must be supplied)

Corresponding to

READ (5, VRF) (COSTI(I), I = 1, NI)

or

READ (5, 1080) (COSTI(I), I = 1, NI)

1080 FORMAT (12F6.0)

in Subroutine ECMP.

Reads COSTI Array depending on the variable format if option 1 is exercised

Otherwise in format (12F6.0) in case of Default.

2.4.11 Data Card(s) 11 (Must be supplied)

Corresponding to

READ (5, VRF) (COSTO(I), I = 1, NO)

or

READ (5, 1080) (COSTO(I), I = 1, NO)

1080 FORMAT (12F6.0)

in subroutine ECMP

Reads COSTO array depending on the variable format if option 1 is exercised.

Otherwise in format (12F6.0) in case of Default.

2.5 List of Programs

2.5.1 User Supplied

1. Main Program : Calls ECMP.
2. SUBROUTINE COMPNT.
3. SUBROUTINE SENSTVTY.

2.5.2 Package

1. ECMP : Calls
 - ADDM ,
 - COMPNT ,
 - IDNTM ,
 - INVM ,
 - NCOMB ,
 - NEWCOM ,
 - PRODM ,
 - SENSTVTY ,
 - STURM ,
 - SUBM ,
 - TRAM ,
 - VEC ,
 - ZERM ,
2. ADDM
3. COMPNT
4. FACTRIAL
5. IDNTM
6. INVM
7. NCOMB : Calls FACTRIAL
8. NEWCOM
9. PRODM
10. SENSTVTY
11. STURM : Uses FUNCTION SQRT from System Tape

- 12. SUBM
- 13. TRAM
- 14. VEC
- 15. ZERM

2.6 Control Cards and Deck Set Up

Control Cards are computer installation dependent; and therefore, the following set provides an example of what may be necessary. The system control cards for the CDC 1604 computer at the Electrical Engineering Department of the Texas Tech University for any problem to be solved by ECMP are shown in Figure 1.

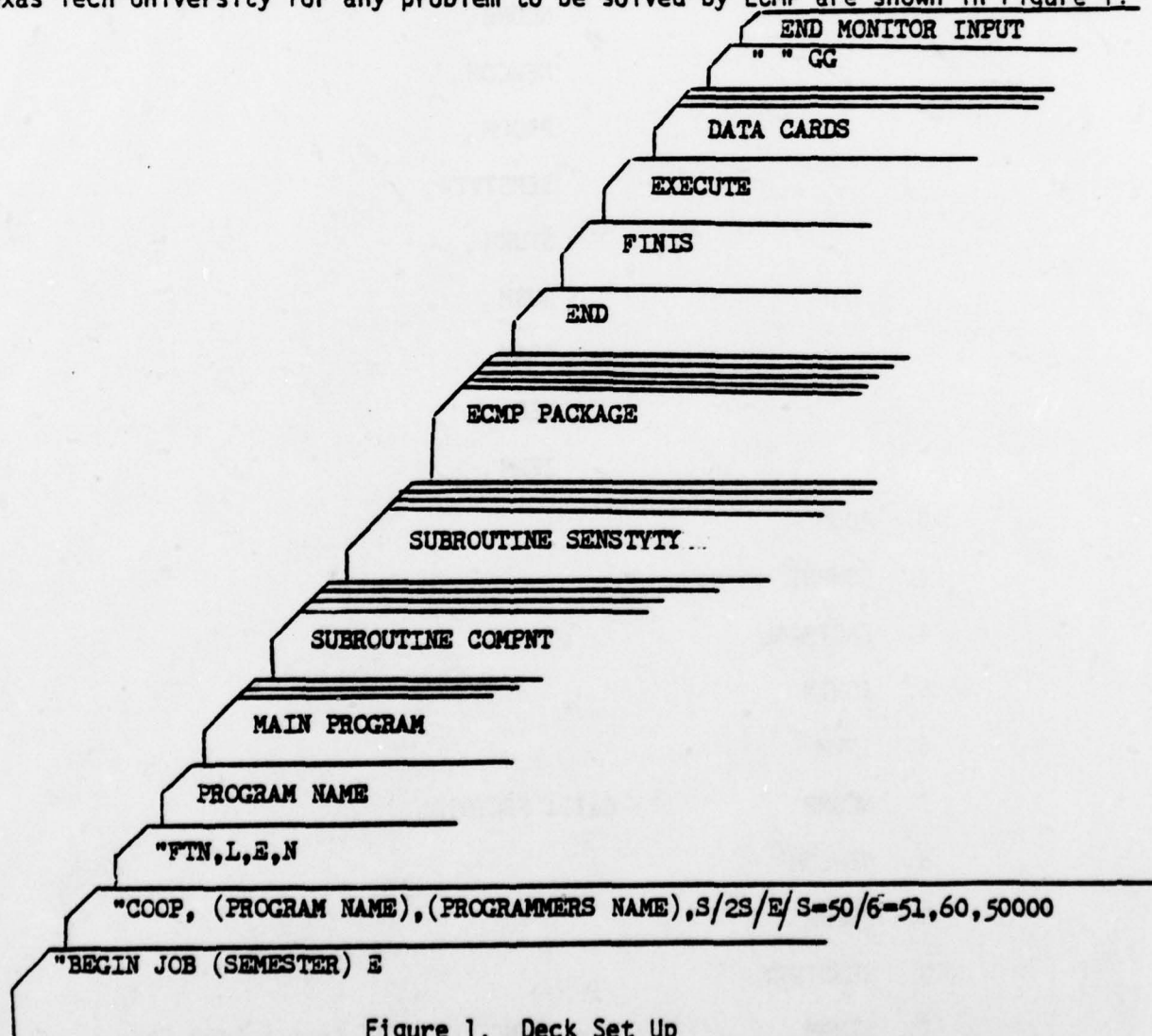


Figure 1. Deck Set Up

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDG

2.7 Program Listing

NO = VARIABLE INDICATING NUMBER OF OUTPUTS
 NP = VARIABLE INDICATING NUMBER OF SENSITIVITY
 PARAMETERS
 NSEQ = VARIABLE INDICATING NUMBER OF ELEMENTS
 IN THE STURM SEQUENCE VECTOR
 NOMB = VECTOR OF SIZE MAXITER TO CONTAIN NUMBER OF
 EIGENVALUES LESS THAN TEST VALUE FOR EACH
 ITERATION
 NVEC = VARIABLE INDICATING NUMBER OF ELEMENTS IN THE
 VECTOR OF INPUTS AND OUTPUTS AT EACH FREQUENCY
 NZ = VARIABLE INDICATING NUMBER OF COMPONENTS
 DDG = VECTOR OF SIZE NP TO CONTAIN ON DIAGONAL
 ELEMENTS OF THE TRIDIAGONAL MATRIX
 P = WORKING MATRIX OF SIZE NO*NZ
 R = WORKING MATRIX OF SIZE NO*NI
 SDG = VECTOR OF SIZE NP TO CONTAIN SUPER OR SUB
 DIAGONAL ELEMENTS OF THE TRIDIAGONAL MATRIX
 SENS = ARRAY OF SIZE NF*NVEC*NP TO CONTAIN
 JACOBIAN MATRIX
 SEQ = VECTOR OF SIZE NSEQ TO CONTAIN STURM SEQUENCE
 SOLUTION FOR A PARTICULAR ITERATION
 SYSH = COMPOSITE SYSTEM TRANSFER FUNCTION
 MATRIX OF SIZE NO*NI
 T = WORKING MATRIX OF SIZE NO*NI
 TCCOF = MATRIX OF SIZE NP*NVEC TO CONTAIN
 TRANSPOSED COEFFICIENTS
 VAL = VECTOR OF SIZE MAXITER TO CONTAIN TEST
 EIGENVALUE
 VRF = VARIABLE OF SIZE 10 TO CONTAIN VARIABLE
 READ FORMAT
 VWF = VARIABLE OF SIZE 10 TO CONTAIN VARIABLE
 WRITE FORMAT
 W = WORKING MATRIX OF SIZE NZ*NI
 XI = IDENTITY MATRIX OF SIZE NZ*NZ
 Z = COMPONENT MATRIX OF SIZE NZ*NZ

ECMP 270
 ECMP 280
 ECMP 290
 ECMP 300
 ECMP 310
 ECMP 320
 ECMP 330
 ECMP 340
 ECMP 350
 ECMP 360
 ECMP 37
 ECMP 380
 ECMP 390
 ECMP 400
 ECMP 410
 ECMP 420
 ECMP 430
 ECMP 440
 ECMP 450
 ECMP 460
 ECMP 470
 ECMP 480
 ECMP 490
 ECMP 500
 ECMP 510
 ECMP 520
 ECMP 530
 ECMP 540
 ECMP 550
 ECMP 560
 ECMP 570
 ECMP 580
 ECMP 590
 ECMP 600
 ECMP 610
 ECMP 620
 ECMP 630
 ECMP 640
 ECMP 650
 ECMP 660
 ECMP 670
 ECMP 680
 ECMP 690
 ECMP 700
 ECMP 710
 ECMP 720
 ECMP 730
 ECMP 740
 ECMP 750
 ECMP 760
 ECMP 770
 ECMP 780
 ECMP 790
 ECMP 800
 ECMP 810
 ECMP 820
 ECMP 830
 ECMP 840
 ECMP 850
 ECMP 860
 ECMP 870
 ECMP 880
 ECMP 890
 ECMP 900
 ECMP 910
 ECMP 920
 ECMP 930
 ECMP 940
 ECMP 950
 ECMP 960
 ECMP 970
 ECMP 980
 ECMP 990
 ECMP1000
 ECMP1010
 ECMP1020
 ECMP1030
 ECMP1040
 ECMP1050
 ECMP1060
 ECMP1070
 ECMP1080
 ECMP1090
 ECMP1100
 ECMP1110
 ECMP1120

THIS PAGE IS BEST QUALITY REPRODUCTION
 FROM COPY FURNISHED TO DDG

REMARKS
 THIS PACKAGE REQUIRES ONE USER SUPPLIED MAIN CALLING
 PROGRAM AND TWO USER SUPPLIED SUBROUTINES COMPNT AND
 SENSVTY IN ADDITION TO UPTO 11 DATA CARDS (OR DATA
 CARD SETS) DEPENDING UPON THE OPTIONS EXERCISED AND
 THE SIZE OF CIRCUIT OR SYSTEM FOR ANALYSIS.
 IT IS STRONGLY URGED THAT THE USER SHOULD STUDY
 FOLLOWING REFERENCE CAREFULLY BEFORE ATTEMPTING
 TO USE THIS PACKAGE FOR ANALYSIS.

EC4P - A SOFTWARE PACKAGE FOR COMPUTATION OF
 MEASURABILITY AND COST TO AID TESTPOINT SELECTION
 BY N. SEN AND R. SAEKS, DEPT OF ELEC ENGR,
 TEXAS TECH UNIVERSITY, LUBBOCK, TX 79409

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED

ADJM, COMPNT, IDNTM, INVM, NCOMB, NEWCOM, PRODM,
 SENSVTY, STURM, SUPM, TRAM, VEC, ZERM

SUBROUTINE ECMP (A,AL11,AL12,AL21,AL22,B,C,CCOF,COF,COL,
 COSTI,COSTO,D,DIFZ,UP,E,F,G,H,INDOPT,LCOMI,LCOMO,MAXCAS,MAXITER,
 NF,NFVEC,NI,NO,NP,NSEQ,NUMH,NVEC,NZ,ODG,P,R,
 S03,SENS,SEQ,SYSM,T,TCCOF,VAL,VRF,VWF,W,XI,Z)

ECMP 10
 ECMP 20
 ECMP 30
 ECMP 40

SUBROUTINE ECMP

THIS PAGE IS BEST QUALITY PRACTICABLE
 FROM COPY FURNISHED TO DDC

PURPOSE

PROGRAM ECMP (ELECTRONIC CIRCUIT MEASURABILITY
 PACKAGE) CALCULATES THE MEASURE DELTA AND THE COST
 INVOLVED FOR EACH POSSIBLE COMBINATION OF THE
 CIRCUIT OR SYSTEM INPUTS AND OUTPUTS

ECMP 50
 ECMP 60
 ECMP 70
 ECMP 80
 ECMP 90
 ECMP 100
 ECMP 110
 ECMP 120
 ECMP 130
 ECMP 140

USAGE

CALL ECMP (A,AL11,AL12,AL21,AL22,B,C,CCOF,COF,COL
 COSTI,COSTO,D,DIFZ,UP,E,F,G,H,INDOPT,LCOMI,LCOMO,MAXCAS,
 MAXITER,NF,NFVEC,NI,NO,NP,NSEQ,NUMH,NVEC,NZ,ODG,
 P,R, S03,SENS,SEQ,SYSM,T,TCCOF,VAL,VRF,VWF,W,XI,Z)

ECMP 150
 ECMP 160
 ECMP 170
 ECMP 180
 ECMP 190
 ECMP 200

DESCRIPTION OF PARAMETERS

A = WORKING MATRIX OF SIZE NZ*NZ
 AL11 = FEEDBACK MATRIX L11 OF SIZE NZ*NZ
 AL12 = POST COMPENSATOR MATRIX L12 OF SIZE NZ*NI
 AL21 = PRE COMPENSATOR MATRIX L21 OF SIZE NO*NZ
 AL22 = FEED FORWARD MATRIX L22 OF SIZE NO*NI
 B = WORKING MATRIX OF SIZE NZ*NZ
 C = WORKING MATRIX OF SIZE NZ*NZ
 CCOF = COEFFICIENT MATRIX OF SIZE NFVEC*NP
 COF = COEFFICIENT ARRAY OF SIZE NF*NVEC*NP
 COL = COLUMN VECTOR OF SIZE NVEC
 COSTI = VECTOR OF SIZE NI TO CONTAIN INPUT COSTS
 COSTO = VECTOR OF SIZE NO TO CONTAIN OUTPUT COSTS
 D = WORKING MATRIX OF SIZE NZ*NZ
 DIFZ = SINGLE PARAMETER SENSITIVE COMPONENT MATRIX
 OF SIZE NZ*NZ
 DP = SYMMETRIC POSITIVE SEMIDEFINITE MATRIX OF
 SIZE NP*NP
 E = WORKING MATRIX OF SIZE NZ*NZ
 F = WORKING MATRIX OF SIZE NZ*NZ
 G = WORKING MATRIX OF SIZE NZ*NZ
 H = WORKING MATRIX OF SIZE NZ*NZ
 INDOPT = VARIABLE IF NON ZERO INTEGER INDICATES SOME
 OPTION EXERCISED BY THE USER
 LCOMI = VECTOR OF SIZE MAXCAS TO CONTAIN
 INPUT COMBINATIONS
 LCOMO = VECTOR OF SIZE MAXCAS TO CONTAIN
 OUTPUT COMBINATIONS
 MAXCAS = VARIABLE INDICATING MAXIMUM NUMBER OF CASES
 MAXITER = VARIABLE INDICATING MAXIMUM NUMBER OF
 STURM SEQUENCE ITERATIONS TO BE PERFORMED
 NF = VARIABLE INDICATING NUMBER OF FREQUENCIES
 NFVEC = VARIABLE INDICATING NUMBER OF ELEMENTS IN THE
 VECTOR OF INPUTS AND OUTPUTS AT ALL FREQUENCIES
 NI = VARIABLE INDICATING NUMBER OF INPUTS

ECMP 210
 ECMP 220
 ECMP 230
 ECMP 240
 ECMP 250
 ECMP 260
 ECMP 270
 ECMP 280
 ECMP 290
 ECMP 300
 ECMP 310
 ECMP 320
 ECMP 330
 ECMP 340
 ECMP 350
 ECMP 360
 ECMP 370
 ECMP 380
 ECMP 390
 ECMP 400
 ECMP 410
 ECMP 420
 ECMP 430
 ECMP 440
 ECMP 450
 ECMP 460
 ECMP 470
 ECMP 480
 ECMP 490
 ECMP 500
 ECMP 510
 ECMP 520
 ECMP 530
 ECMP 540
 ECMP 550
 ECMP 560


```

      READ (5,VWF)  (COSTI(I),I=1,N1)
      READ (5,VWF)  (COSTO(I),I=1,N0)
      GO TO 140
120  WRITE (6,1070)
      SFREQ = 1.0
      STEP = 1.0

```

OR WITH INTERNAL FORMAT IF OPTION NOT EXERCISED

```

150  READ (5,1080)  ((AL11(I,J),J=1,NZ),I=1,NZ)
      READ (5,1080)  ((AL12(I,J),J=1,N1),I=1,NZ)
      READ (5,1080)  ((AL21(I,J),J=1,NZ),I=1,N0)
      READ (5,1080)  ((AL22(I,J),J=1,N1),I=1,N0)
      READ (5,1080)  (COSTI(I),I=1,N1)
      READ (5,1080)  (COSTO(I),I=1,N0)

```

PRINT INPUT DATA IN APPROPRIATE FORMAT IF OPTION(S) EXERCISED

```

140  IF (INDOPT.NE.0) 150,190

```

```

150  IF (INDX4.NE.0) 160,190

```

```

160  WRITE (6,1090)
      WRITE (6,1100)  NZ
      WRITE (6,1110)  N1
      WRITE (6,1120)  N0
      WRITE (6,1130)  NP
      WRITE (6,1140)  NF
      WRITE (6,1150)  NVEC
      WRITE (6,1160)  NFVEC
      WRITE (6,1170)  NSEQ
      WRITE (6,1180)  MAXCAS
      WRITE (6,1190)  MAXITER
      WRITE (6,1200)  SFREQ
      WRITE (6,1210)  STEP

```

```

      IF (INDX2.NE.0) 170,180

```

```

170  WRITE (6,1220)
      WRITE (6,VWF)  ((AL11(I,J),J=1,NZ),I=1,NZ)
      WRITE (6,1230)
      WRITE (6,VWF)  ((AL12(I,J),J=1,N1),I=1,N1)
      WRITE (6,1240)
      WRITE (6,VWF)  ((AL21(I,J),J=1,NZ),I=1,N0)
      WRITE (6,1250)
      WRITE (6,VWF)  ((AL22(I,J),J=1,N1),I=1,N0)
      WRITE (6,1260)
      WRITE (6,VWF)  (COSTI(I),I=1,N1)
      WRITE (6,1270)
      WRITE (6,VWF)  (COSTO(I),I=1,N0)
      GO TO 190

```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```

180  WRITE (6,1220)
      WRITE (6,1280)  ((AL11(I,J),J=1,NZ),I=1,NZ)
      WRITE (6,1230)
      WRITE (6,1280)  ((AL12(I,J),J=1,N1),I=1,N1)
      WRITE (6,1240)
      WRITE (6,1280)  ((AL21(I,J),J=1,NZ),I=1,N0)
      WRITE (6,1250)

```

ECMP169
ECMP170
ECMP171
ECMP172
ECMP173
ECMP174
ECMP175
ECMP176
ECMP177
ECMP178
ECMP179
ECMP180
ECMP181
ECMP182
ECMP183
ECMP184
ECMP185
ECMP186
ECMP187
ECMP188
ECMP189
ECMP190
ECMP191
ECMP192
ECMP193
ECMP194
ECMP195
ECMP196
ECMP197
ECMP198
ECMP199
ECMP200
ECMP201
ECMP202
ECMP203
ECMP204
ECMP205
ECMP206
ECMP207
ECMP208
ECMP209
ECMP210
ECMP211
ECMP212
ECMP213
ECMP214
ECMP215
ECMP216
ECMP217
ECMP218
ECMP219
ECMP220
ECMP221
ECMP222
ECMP223
ECMP224

NOJMB FURTHER REQUIRES FACTRIAL AND
STORM FURTHER REQUIRES SORT

DATE WRITTEN

AUGUST 1977

DIMENSION A(NZ,NZ),AL11(NZ,NZ),AL12(NZ,N1),AL21(N0,NZ),
AL22(N0,N1),F(NZ,NZ),C(NZ,NZ),COOF(NFVEC,NP),COF(NF,NVEC,NP),
COL(NVEC),COSTI(N1),COSTO(N0),D(NZ,NZ),DIFZ(NZ,NZ),UP(NP,NP),
F(NZ,N7),F(N7,NZ),G(NZ,N7),H(N7,NZ),LCOMI(MAXCAS),LCOMO(MAXCAS),
INDX3(MAXITER),JDG(NP),P(N0,NZ),P(N0,N1),SDG(NP),SFNS(NF,NVEC,NP),
SEI(NSEI),SYSM(N0,N1),T(N0,N1),TCCOF(NP,NFVEC),VAL(MAXITER),
VRF(10),VWF(10),X(NZ,N1),XI(NZ,NZ),Z(NZ,NZ)

CHECK IF ANY OPTION(S) EXERCISED BY USER

WRITE (6,1000)
IF (INDOPT.NE.0) 10,120
10 READ (5,1010) INDX1,INDX2,INDX3,INDX4,INDX5,INDX6,INDX7,
INDX8,INDX9,INDX10,INDX11,INDX12

CHECK IF VARIABLE READ FORMAT DESIRED

IF (INDX1.NE.0) 20,30
20 READ (5,1020) (VRF(I),I=1,10)
GO TO 40
30 WRITE (6,1030)

CHECK IF VARIABLE WRITE FORMAT DESIRED

40 IF (INDX2.NE.0) 50,60
50 READ (5,1020) (VWF(I),I=1,10)
GO TO 70
60 WRITE (6,1040)

CHECK IF DIFFERENT START FREQUENCY AND STEP SIZE DESIRED

70 IF (INDX3.NE.0) 80,90
80 READ (5,1050) SFREQ,STEP
GO TO 100
90 WRITE (6,1060)
SFREQ = 1.0
STEP = 1.0

READ CONNECTION MATRICES AND COST INFORMATION

100 IF (INDX1.NE.0) 110,130

WITH VARIABLE READ FORMAT IF OPTION EXERCISED

110 READ (5,VRF) ((AL11(I,J),J=1,NZ),I=1,NZ)
READ (5,VRF) ((AL12(I,J),J=1,N1),I=1,NZ)
READ (5,VRF) ((AL21(I,J),J=1,NZ),I=1,N0)
READ (5,VRF) ((AL22(I,J),J=1,N1),I=1,N0)

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

C	COMPUTATION OF THE JACOBIAN MATRIX	ECMP2810
C		ECMP2820
350	CALL PRODM (AL11,NZ,NZ,D,NZ,N7,E,NZ,NZ)	ECMP2830
	CALL ADDM (E,NZ,NZ,XI,NZ,N7,F,NZ,NZ)	ECMP2840
	CALL PRODM (C,N7,NZ,DIF7,NZ,N7,G,NZ,NZ)	ECMP2850
	CALL PRODM (G,NZ,NZ,F,NZ,NZ,H,NZ,NZ)	ECMP2860
	CALL PRODM (AL21,NO,NZ,H,NZ,NZ,P,NO,NZ)	ECMP2870
	CALL PRODM (P,NO,NZ,AL12,NZ,NI,R,NO,NI)	ECMP2880
	CALL VEC (R,NO,NI,COL,NVEC)	ECMP2890
C		ECMP2900
	DO 360 JS = 1,NVEC	ECMP2910
	SENS (IS,JS,KS) = COL (JS)	ECMP2920
360	CONTINUE	ECMP2930
C		ECMP2940
	KS = KS + 1	ECMP2950
	IF (KS.LE.NP) 300,370	ECMP2960
370	S = S + STEP	ECMP2970
	IS = IS + 1	ECMP2980
	IF (IS.LE.NF) 200,380	ECMP2990
C		ECMP3000
C	OPTIONAL PRINTOUT OF THE JACOBIAN MATRIX	ECMP3010
C		ECMP3020
380	IF (INDOPT.NE.0) 390,430	ECMP3030
390	IF (INDX8.NE.0) 400,430	ECMP3040
400	WRITE (6,1320)	ECMP3050
	IF (INDX2.NE.0) 410,420	ECMP3060
410	WRITE (6,VWF) (((SENS(I,J,K),K=1,NP),J=1,NVEC),I=1,NF)	ECMP3070
	GO TO 430	ECMP3080
420	WRITE (6,1280) (((SENS(I,J,K),K=1,NP),J=1,NVEC),I=1,NF)	ECMP3090
430	NCASE = 1	ECMP3100
C		ECMP3110
C	SELECT THE NUMBER OF INPUTS TO BE CONSIDERED	ECMP3120
C		ECMP3130
	DO 610 I = 1,NI	ECMP3140
C		ECMP3150
C	INITIALIZE INPUT COMBINATION ARRAY TO TOTAL NUMBER OF INPUTS	ECMP3160
C		ECMP3170
	DO 440 I1 = 1,I	ECMP3180
	LCOMI (I1) = NI	ECMP3190
440	CONTINUE	ECMP3200
C		ECMP3210
C	FIND MAXIMUM NUMBER OF COMBINATIONS POSSIBLE FOR SELECTED	ECMP3220
C	NUMBER OF INPUTS	ECMP3230
C		ECMP3240
	CALL NCOMB (NCR,I,NI)	ECMP3250
C		ECMP3260
C	GET ONE SUCH COMBINATION OF INPUTS AT A TIME	ECMP3270
C		ECMP3280
	DO 610 J = 1,NCR	ECMP3290
	CALL NEWCOM (LCOMI,I,NI)	ECMP3300
C		ECMP3310
C	SELECT THE NUMBER OF OUTPUTS TO BE CONSIDERED	ECMP3320
C		ECMP3330
	DO 610 K = 1,NO	ECMP3340
C		ECMP3350
C	INITIALIZE OUTPUT COMBINATION ARRAY TO TOTAL NUMBER OF OUTPUTS	ECMP3360

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC


```

WRITE (6,1260) ((AL22(I,J),J=1,N1),I=1,N0)
WRITE (6,1260)
WRITE (6,1280) (COSTI(I),I=1,N1)
WRITE (6,1270)
WRITE (6,1280) (COSTO(I),I=1,N0)

```

```

190 IS = 1
S = SFREQ

```

```

200 KS = 1
CALL ZFRM(Z,NZ,NZ)
CALL COMPNT (NZ,S,Z)

```

OPTIONAL PRINTOUT OF THE COMPONENT MATRIX Z

```

IF (INDOPT.NE.0) 210,250
210 IF (INDX5.NE.0) 220,250
220 WRITE (6,1290) S
IF (INDX2.NE.0) 230,240
230 WRITE (6,VWF) ((Z(I,J),J=1,NZ),I=1,NZ)
GO TO 250
240 WRITE (6,1280) ((Z(I,J),J=1,NZ),I=1,NZ)

```

COMPUTATION OF THE COMPOSITE SYSTEM TRANSFER FUNCTION

```

250 CALL PRODM (Z,NZ,NZ,AL11,NZ,NZ,A,NZ,NZ)
CALL IDNTM (XI,NZ,NZ)
CALL SUBM (XI,NZ,NZ,A,NZ,NZ,B,NZ,NZ)
CALL INVM (B,NZ,NZ,C,NZ,NZ)
CALL PRODM (C,NZ,NZ,Z,NZ,NZ,D,NZ,NZ)
CALL PRODM (D,NZ,NZ,AL12,NZ,N1,W,NZ,N1)
CALL PRODM (AL21,N0,NZ,W,NZ,N1,T,N0,N1)
CALL ADDM (AL22,N0,N1,T,N0,N1,SYSM,N0,N1)

```

OPTIONAL PRINTOUT OF THE COMPOSITE SYSTEM TRANSFER FUNCTION

```

IF (INDOPT.NE.0) 260,300
260 IF (INDX6.NE.0) 270,300
270 WRITE (6,1300) S
IF (INDX2.NE.0) 280,290
280 WRITE (6,VWF) ((SYSM(I,J),J=1,N1),I=1,N0)
GO TO 300
290 WRITE (6,1280) ((SYSM(I,J),J=1,N1),I=1,N0)

```

```

300 CALL ZFRM (DIFZ,NZ,NZ)
CALL SENSTVTY (DIFZ,KS,NZ,S)

```

OPTIONAL PRINTOUT OF THE DIFZ MATRIX

```

IF (INDOPT.NE.0) 310,350
310 IF (INDX7.NE.0) 320,350
320 WRITE (6,1310) KS,S
IF (INDX2.NE.0) 330,340
330 WRITE (6,VWF) ((DIFZ(I,J),J=1,NZ),I=1,NZ)
GO TO 350
340 WRITE (6,1280) ((DIFZ(I,J),J=1,NZ),I=1,NZ)

```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDG

ECMP2250
ECMP2260
ECMP2270
ECMP2280
ECMP2290
ECMP2300
ECMP2310
ECMP2320
ECMP2330
ECMP2340
ECMP2350
ECMP2360
ECMP2370
ECMP2380
ECMP2390
ECMP2400
ECMP2410
ECMP2420
ECMP2430
ECMP2440
ECMP2450
ECMP2460
ECMP2470
ECMP2480
ECMP2490
ECMP2500
ECMP2510
ECMP2520
ECMP2530
ECMP2540
ECMP2550
ECMP2560
ECMP2570
ECMP2580
ECMP2590
ECMP2600
ECMP2610
ECMP2620
ECMP2630
ECMP2640
ECMP2650
ECMP2660
ECMP2670
ECMP2680
ECMP2690
ECMP2700
ECMP2710
ECMP2720
ECMP2730
ECMP2740
ECMP2750
ECMP2760
ECMP2770
ECMP2780
ECMP2790
ECMP2800


```

0
DO 450 K1 = 1,K
  LCOMO (K1) = NO
450 CONTINUE

      FIND MAXIMUM NUMBER OF COMBINATIONS POSSIBLE FOR SELECTED
      NUMBER OF OUTPUTS

      CALL NCOMB (KCR,K,NO)

      GET ONE SUCH COMBINATION OF OUTPUTS AT A TIME

      DO 510 L = 1,KCR
      CALL NENCOM (LCOMO,K,NO)

      INITIALIZE COF ARRAY TO ZERO

      DO 460 IZ = 1,NF
      DO 460 JZ = 1,NVEC
      DO 460 KZ = 1,NP
      COF (IZ,JZ,KZ) = 0.0
460 CONTINUE

      INITIALIZE CCOF MATRIX TO ZERO

      DO 470 IZ = 1,NFVEC
      DO 470 JZ = 1,NP
      CCOF (IZ,JZ) = 0.0
470 CONTINUE

      FOR SELECTED COMBINATION OF INPUTS AND OUTPUTS FORM
      THE COF ARRAY FROM OVERALL JACOBIAN

      DO 480 IA = 1,NF
      DO 480 JA = 1,I
      DO 480 KA = 1,K
      DO 480 LA = 1,NP
      JKA = JA + KA
      JF = LCOMI (JA)
      KF = LCOMO (KA)
      JKF = (JF-1) * NO + KF
      COF (IA,JKA,LA) = SENS (IA,JKF,LA)
480 CONTINUE

      COMPUTE COST FOR THE SELECTED COMBINATION OF INPUTS AND OUTPUTS

      CCOST = 0.0
      DO 490 JA = 1,I
      JF = LCOMI (JA)
      CCOST = CCOST + COSTI (JF)
490 CONTINUE

      DO 500 KA = 1,K
      KF = LCOMO (KA)
      CCOST = CCOST + COSTO (KF)
500 CONTINUE

```

```

ECMP3370
ECMP3380
ECMP3390
ECMP3400
ECMP3410
ECMP3420
ECMP3430
ECMP3440
ECMP3450
ECMP3460
ECMP3470
ECMP3480
ECMP3490
ECMP3500
ECMP3510
ECMP3520
ECMP3530
ECMP3540
ECMP3550
ECMP3560
ECMP3570
ECMP3580
ECMP3590
ECMP3600
ECMP3610
ECMP3620
ECMP3630
ECMP3640
ECMP3650
ECMP3660
ECMP3670
ECMP3680
ECMP3690
ECMP3700
ECMP3710
ECMP3720
ECMP3730
ECMP3740
ECMP3750
ECMP3760
ECMP3770
ECMP3780
ECMP3790
ECMP3800
ECMP3810
ECMP3820
ECMP3830
ECMP3840
ECMP3850
ECMP3860
ECMP3870
ECMP3880
ECMP3890
ECMP3900
ECMP3910
ECMP3920

```

CONVERT THREE DIMENSIONAL COF ARRAY INTO CCOF MATRIX

IK = NFVEC

IC = 1

N1 = 0

N2 = 1

N2 = IK

510 DO 520 JM = N1,N2

DO 520 KC = 1,NP

JC = JM - N

CCOF (JM,KC) = COF (IC,JC,KC)

520 CONTINUE

IC = IC + 1

N1 = N1 + IK

N2 = N2 + IK

N2 = N2 + IK

IF (IC.LE.NF) 510,530

OPTIONAL PRINTOUT OF THE COEFFICIENT MATRIX CCOF

530 IF (INDOPT.NE.0) 540,580

540 IF (INDX9.NE.0) 550,580

550 WRITE (6,1330) NCASE

IF (INDX2.NE.0) 560,570

560 WRITE (6,VWF) ((CCOF(IW,JW),JW=1,NP),IW=1,NFVEC)

GO TO 580

570 WRITE (6,1280) ((CCOF(IW,JW),JW=1,NP),IW=1,NFVEC)

COMPUTE SYMMETRIC POSITIVE SEMIDEFINITE MATRIX DP

580 CALL TRAM (CCOF,NFVEC,NP,TCCOF,NP,NFVEC)

CALL PRODM (TCCOF,NP,NFVEC,CCOF,NFVEC,NP,DP,NP,NP)

OPTIONAL PRINTOUT OF THE DP MATRIX

IF (INDOPT.NE.0) 590,630

590 IF (INDX10.NE.0) 600,630

600 WRITE (6,1340) NCASE

IF (INDX2.NE.0) 610,620

610 WRITE (6,VWF) ((DP(IW,JW),JW=1,NP),IW=1,NP)

GO TO 630

620 WRITE (6,1280) ((DP(IW,JW),JW=1,NP),IW=1,NP)

COMPUTE DELTA BY STURM SEQUENCE SOLUTION

630 CALL STURM (DP,ICT,INDOPT,INDX11,INDX12,MAXITER,NCASE,
-NP,ISEQ,NUMB,ODG,SDG,SEQ,VAL,VALUE)

PRINTOUT OF THE RESULTS ACCOMMODATING CHANGES REQUIRED
DUE TO OPTIONAL PRINTOUTS, IF ANY

IF (INDOPT.NE.0) 640,650

640 IF ((INDX9.NE.0).OR.(INDX10.NE.0).OR.(INDX11.NE.0).OR.

ECMP3930

ECMP3940

ECMP3950

ECMP3960

ECMP3970

ECMP3980

ECMP3990

ECMP4000

ECMP4010

ECMP4020

ECMP4030

ECMP4040

ECMP4050

ECMP4060

ECMP4070

ECMP4080

ECMP4090

ECMP4100

ECMP4110

ECMP4120

ECMP4130

ECMP4140

ECMP4150

ECMP4160

ECMP4170

ECMP4180

ECMP4190

ECMP4200

ECMP4210

ECMP4220

ECMP4230

ECMP4240

ECMP4250

ECMP4260

ECMP4270

ECMP4280

ECMP4290

ECMP4300

ECMP4310

ECMP4320

ECMP4330

ECMP4340

ECMP4350

ECMP4360

ECMP4370

ECMP4380

ECMP4390

ECMP4400

ECMP4410

ECMP4420

ECMP4430

ECMP4440

ECMP4450

ECMP4460

ECMP4470

ECMP4480

650	IF (1.EQ.1) 660,670	ECMP4490
660	WRITE (6,1350)	ECMP4500
670	WRITE (6,1360) NCASE,LCOMI(1),LCOMO(1),CCOST,ICT,VALUE	ECMP4510
	IF (1.EQ.K) 680,710	ECMP4520
680	IF (1.EQ.1) 690,800	ECMP4530
C		ECMP4540
690	DO 720 IW = 2,I	ECMP4550
	WRITE (6,1370) LCOMI(IW),LCOMO(IW)	ECMP4560
700	CONTINUE	ECMP4570
	GO TO 800	ECMP4580
C		ECMP4590
710	IF (1.EQ.1) 720,730	ECMP4600
720	WRITE (6,1380) (LCOMI(IW),IW = 2,I)	ECMP4610
	GO TO 800	ECMP4620
C		ECMP4630
730	IF (1.EQ.1) 740,750	ECMP4640
740	WRITE (6,1390) (LCOMO(IW),IW = 2,K)	ECMP4650
	GO TO 800	ECMP4660
C		ECMP4670
750	IF (1.GT.K) 760,780	ECMP4680
760	DO 770 IW = 2,K	ECMP4690
	WRITE (6,1370) LCOMI(IW),LCOMO(IW)	ECMP4700
770	CONTINUE	ECMP4710
	WRITE (6,1380) (LCOMI(IW),IW = K,I)	ECMP4720
	GO TO 800	ECMP4730
C		ECMP4740
780	DO 790 IW = 2,I	ECMP4750
	WRITE (6,1370) LCOMI(IW),LCOMO(IW)	ECMP4760
790	CONTINUE	ECMP4770
	WRITE (6,1390) (LCOMO(IW),IW = 1,K)	ECMP4780
C		ECMP4790
GO TO NEXT CASE AND RESTART FOR A NEW COMBINATION		ECMP4800
OF INPUTS AND OUTPUTS		ECMP4810
C		ECMP4820
800	NCASE = NCASE + 1	ECMP4830
810	CONTINUE	ECMP4840
C		ECMP4850
FORMAT STATEMENTS		ECMP4860
C		ECMP4870
1000	FORMAT (1H1)	ECMP4880
1010	FORMAT (12I5)	ECMP4890
1020	FORMAT (10A8)	ECMP4900
1030	FORMAT (62H0 OPTION 1 NOT EXERCISED, DEFAULT FORMAT ON INPUT DATA	ECMP4910
	- ASSUMED)	ECMP4920
1040	FORMAT (63H0 OPTION 2 NOT EXERCISED, DEFAULT FORMAT ON OUTPUT DATA	ECMP4930
	- ASSUMED)	ECMP4940
1050	FORMAT (2F6.0)	ECMP4950
1060	FORMAT (72H0 OPTION 3 NOT EXERCISED, DEFAULT VALUES FOR SFREQ AND	ECMP4960
	- STEP SIZE ASSUMED)	ECMP4970
1070	FORMAT (77H0 NO OPTIONS EXERCISED, DEFAULT VALUES ASSUMED, NORMAL	ECMP4980
	- PRINTOUT WILL FOLLOW)	ECMP4990
1080	FORMAT (12F6.0)	ECMP5000
1090	FORMAT (50H1 A PRINTOUT OF THE NUMERICAL DATA READ IN FOLLOWS////)	ECMP5010
1100	FORMAT (1H0,10X,42#NUMBER OF COMPONENTS	ECMP5020
	NZ = ,13)	ECMP5030
1110	FORMAT (1H0,10X,42#NUMBER OF INPUTS	ECMP5040
	NI = ,13)	


```

1120 FORMAT (1H0,10X,42HNUMBER OF OUTPUTS NO = ,I3) ECMP5050
1130 FORMAT (1H0,10X,42HNUMBER OF SENSITIVITY PARAMETERS NP = ,I3) ECMP5060
1140 FORMAT (1H0,10X,42HNUMBER OF FREQUENCIES FOR ANALYSIS NF = ,I3) ECMP5070
1150 FORMAT (1H0,10X,42HLENGTH OF VECTOR FOR EACH FREQ NVEC = ,I3) ECMP5080
1160 FORMAT (1H0,10X,42HLENGTH OF VECTOR FOR ALL FREQS NFVEC = ,I3) ECMP5090
1170 FORMAT (1H0,10X,42HNUMBER OF ELEMENTS IN STORM SEQ NSEQ = ,I3) ECMP5100
1180 FORMAT (1H0,10X,42HNUMBER OF CASES TO BE ANALYZED MAXCAS = ,I3) ECMP5110
1190 FORMAT (1H0,10X,42HMAX NUMBER OF STORM SEQ ITER MAXITER = ,I3) ECMP5120
1200 FORMAT (1H0,10X,41HSTARTING FREQUENCY SFREQ = ,F6.2) ECMP5130
1210 FORMAT (1H0,10X,41HSTEP SIZE FOR FREQUENCY STEP = ,F6.2) ECMP5140
1220 FORMAT (19H1 THE L11 MATRIX IS//) ECMP5150
1230 FORMAT (19H1 THE L12 MATRIX IS//) ECMP5160
1240 FORMAT (19H1 THE L21 MATRIX IS//) ECMP5170
1250 FORMAT (19H1 THE L22 MATRIX IS//) ECMP5180
1260 FORMAT (23H1 THE COST OF INPUTS IS//) ECMP5190
1270 FORMAT (24H1 THE COST OF OUTPUTS IS//) ECMP5200
1280 FORMAT (1H0,6E2).3) ECMP5210
1290 FORMAT (42H1 THE COMPONENT MATRIX Z AT A FREQUENCY OF,2X,F6.2,2X, ECMP5220
+2HIS//) ECMP5230
1300 FORMAT (70H1 THE COMPOSITE SYSTEM TRANSFER FUNCTION MATRIX SYSM AT ECMP5240
+ A FREQUENCY OF,2X,F6.2,2X,2HIS//) ECMP5250
1310 FORMAT (43H1 THE DIF7 MATRIX FOR SENSITIVITY PARAMETER,2X,I3, ECMP5260
+2X,17HAT A FREQUENCY OF,2X,F6.2,2X,2HIS//) ECMP5270
1320 FORMAT (24H1 THE JACOBIAN MATRIX IS//) ECMP5280
1330 FORMAT (32H1 THE COEFFICIENT MATRIX IN CASE,2X,I3,2X,2HIS//) ECMP5290
1340 FORMAT (23H1 THE DP MATRIX IN CASE,2X,I3,2X,2HIS//) ECMP5300
1350 FORMAT (1H1,5X,4HCASE,15X,6HINPUTS,15X,7HOUTPUTS,15X, ECMP5310
+4HCOST,17X,5HDELTA,5X,20HLAST TEST EIGENVALUE) ECMP5320
1360 FORMAT (1H0,5X,14,15X,I3,20X,I3,5X,E20.3,15X,I3,5X,E20.3) ECMP5330
1370 FORMAT (1H0,24X,I3,20X,I3) ECMP5340
1380 FORMAT (1H0,24X,I3) ECMP5350
1390 FORMAT (1H0,47X,I3) ECMP5360
C ECMP5370
RETURN ECMP5380
END ECMP5390

```


SUBROUTINE ADDM (X,IX,JX,Y,IY,JY,Z,IZ,JZ)

ADDM 10
ADDM 20

SUBROUTINE ADDM

ADDM 30
ADDM 40
ADDM 50

PURPOSE

ADDM 60
ADDM 70

ADDS TWO MATRICES X AND Y TO PRODUCE MATRIX Z

ADDM 80
ADDM 90

USAGE

CALL ADDM (X,IX,JX,Y,IY,JY,Z,IZ,JZ)

ADDM 100
ADDM 110
ADDM 120

DESCRIPTION OF PARAMETERS

ADDM 130
ADDM 140

X = FIRST MATRIX TO BE ADDED

IX AND JX = DIMENSIONS OF MATRIX X

ADDM 150

Y = SECOND MATRIX TO BE ADDED

ADDM 160

IY AND JY = DIMENSIONS OF MATRIX Y

ADDM 170

Z = RESULTING SUM MATRIX

ADDM 180

IZ AND JZ = DIMENSIONS OF MATRIX Z

ADDM 190

REMARKS

ADDM 200
ADDM 210

MATRICES MUST BE PROPERLY DIMENSIONED BY CALLING PROGRAM

ADDM 220

TERMINATES JOB IF DIMENSIONS DO NOT MATCH

ADDM 230

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED

ADDM 240

NONE

ADDM 250

ADDM 260

ADDM 270

DIMENSION X (IX,JX), Y (IY,JY), Z (IX,JX)

ADDM 280

IF (IX.EQ.IY) 10,40

ADDM 290

10 IF (JX.EQ.JY) 20,40

ADDM 300

20 IZ = IX

ADDM 310

JZ = JX

ADDM 320

ADDM 330

ADDM 340

DO 30 I = 1,IZ

ADDM 350

DO 30 J = 1,JZ

ADDM 360

Z(I,J) = X(I,J) + Y(I,J)

ADDM 370

30 CONTINUE

ADDM 380

RETURN

ADDM 390

ADDM 400

ADDM 410

ADDM 420

40 WRITE (6,1000)

1000 FORMAT (47H **ERROR** MATRICES NOT COMPATIBLE FOR ADDITION)

ADDM 430

ADDM 440

END

ADDM 450

SUBROUTINE FACTRIAL (IFACT,N)

FACT 10

FACT 20

FACT 30

FACT 40

SUBROUTINE FACTRIAL

FACT 50

FACT 60

PURPOSE

FACT 70

FACT 80

CALCULATES FACTORIAL OF A NUMBER N

FACT 90

FACT 100

USAGE

CALL FACTRIAL (IFACT,N)

FACT 110

FACT 120

DESCRIPTION OF PARAMETERS

FACT 130

IFACT = VALUE OF FACTORIAL

FACT 140

N = NUMBER WHOSE FACTORIAL IS DESIRED

FACT 150

FACT 160

REMARKS

FACT 170

NONE

FACT 180

FACT 190

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED

FACT 200

NONE

FACT 210

FACT 220

FACT 230

IFACT = 1

FACT 240

FACT 250

IF (N.NE.0) 10,30

FACT 260

FACT 270

10 I = 1

FACT 280

FACT 290

DO 20 J = 1,N

IFACT = IFACT * I

FACT 300

FACT 310

20 I = I + 1

FACT 320

FACT 330

30 RETURN

FACT 340

END

SC-Routine IDNTM(X,IX,JX)

IDNT 10

IDNT 20

IDNT 30

IDNT 40

SUBROUTINE IDNTM

IDNT 50

IDNT 60

PURPOSE

IDNT 70

PRODUCES AN IDENTITY MATRIX

IDNT 80

IDNT 90

USAGE

IDNT 100

CALL IDNTM(X,IX,JX)

IDNT 110

IDNT 120

DESCRIPTION OF PARAMETERS

IDNT 130

X = RESULTING IDENTITY MATRIX

IDNT 140

IX AND JX = DIMENSIONS OF MATRIX X

IDNT 150

IDNT 160

REMARKS

IDNT 170

MATRIX MUST BE PROPERLY DIMENSIONED BY CALLING PROGRAM

IDNT 180

TERMINATES JOB IF MATRIX NOT SQUARE

IDNT 190

IDNT 200

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED

IDNT 210

NONE

IDNT 220

IDNT 230

IDNT 240

IDNT 250

IDNT 260

DIMENSION X (IX,JX)

IDNT 270

IF (IX.EQ.JX) 10,40

IDNT 280

10 DO 20 I = 1,IX

IDNT 290

DO 20 J = 1,JX

IDNT 300

X(I,J) = 0.0

IDNT 310

20 CONTINUE

IDNT 320

IDNT 330

DO 30 K = 1,IX

IDNT 340

X(K,K) = 1.0

IDNT 350

30 CONTINUE

IDNT 360

RETURN

IDNT 370

IDNT 380

40 WRITE (6,1000)

IDNT 390

1000 FORMAT (41H **ERROR** IDENTITY MATRIX MUST BE SQUARE)

IDNT 400

IDNT 410

END

IDNT 420

----- INVM (X,IX,JX,Y,IY,JY)

INVM 10

INVM 20

----- INVM 30

INVM 40

SUBROUTINE INVM

INVM 50

PURPOSE

INVM 60

COMPUTES INVERSE OF MATRIX X

INVM 70

INVM 80

USAGE

INVM 90

CALL INVM (X,IX,JX,Y,IY,JY)

INVM 100

INVM 110

DESCRIPTION OF PARAMETERS

INVM 120

X = MATRIX WHOSE INVERSE IS TO BE COMPUTED

INVM 130

IX AND JX = DIMENSIONS OF MATRIX X

INVM 140

Y = RESULTING MATRIX WHICH IS INVERSE OF X

INVM 150

IY AND JY = DIMENSIONS OF MATRIX Y

INVM 160

TRAM 170

REMARKS

INVM 180

MATRICES MUST BE PROPERLY DIMENSIONED BY CALLING PROGRAM

INVM 190

TERMINATES JOB IF DIMENSIONS DO NOT MATCH

INVM 200

LEAVES GARBAGE IN MATRIX X ON EXIT

INVM 210

INVM 220

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED

INVM 230

NONE

INVM 240

INVM 250

METHOD

INVM 260

USES GAUSSIAN ELIMINATION METHOD

INVM 270

INVM 280

INVM 290

----- INVM 300

DIMENSION X(IX,JX) , Y(IX,JX)

INVM 310

IF (IX.EQ.JX) 10,170

INVM 320

INVM 330

INVM 340

INVM 350

INVM 360

INVM 370

INVM 380

INVM 390

INVM 400

INVM 410

INVM 420

INVM 430

INVM 440

INVM 450

INVM 460

INVM 470

INVM 480

INVM 490

INVM 500

INVM 510

INVM 520

INVM 530

INVM 540

INVM 550

INVM 560

10 DO 40 I = 1,IX

20 DO 40 J = 1,IX

IF (J.EQ.I) 20,30

20 Y(I,J) = 1.0

30 DO 40 40

30 Y(I,J) = 0.0

40 CONTINUE

DO 60 I = 1,IX

IF (X(I,I).NE.1.0) 90,50

50 DO 60 JJ = 1,IX

IF (JJ.EQ.I) 80,60

60 IF (X(JJ,I).EQ.0.0) 80,70

70 C = X(JJ,I)

DO 60 J = 1,IX

Y(JJ,J) = Y(JJ,J) - C*Y(I,J)

X(JJ,J) = X(JJ,J) - C*X(I,J)

80 CONTINUE

DO 10 160

90 IF (X(I,I).EQ.0.0) 120,100

100 R = X(I,I)


```

C
    GO 110 J = 1, IX
    Y(I,J) = Y(I,J) / R
    X(I,J) = X(I,J) / R
110 CONTINUE
C
    GO TO 50
120 K = I + 1
C
    GO 130 KK = K, IX
    IF (X(KK,I) .EQ. 0.0) 130,140
130 CONTINUE
    GO TO 170
C
140 GO 150 J = 1, IX
    E = Y(I,J)
    Y(I,J) = Y(KK,J)
    Y(KK,J) = E
    D = X(I,J)
    X(I,J) = X(KK,J)
    X(KK,J) = D
150 CONTINUE
C
    GO TO 40
160 IY = IX
    JY = JX
    RETURN
C
170 WRITE (5,1000)
1000 FORMAT (37H **ERROR** THIS MATRIX HAS NO INVERSE)
    END

```

```

INVM 57
INVM 58
INVM 59
INVM 60
INVM 61
INVM 62
INVM 63
INVM 64
INVM 65
INVM 66
INVM 67
INVM 68
INVM 69
INVM 70
INVM 71
INVM 72
INVM 73
INVM 74
INVM 75
INVM 76
INVM 77
INVM 78
INVM 79
INVM 80
INVM 81
INVM 82
INVM 83
INVM 84
INVM 85
INVM 86
INVM 87

```

SUBROUTINE NCOMB (NC,NR,NT)

NCOM 1

NCOM 2

NCOM 3

NCOM 4

NCOM 5

NCOM 6

NCOM 7

NCOM 8

NCOM 9

NCOM 10

NCOM 11

NCOM 12

NCOM 13

NCOM 14

NCOM 15

NCOM 16

NCOM 17

NCOM 18

NCOM 19

NCOM 20

NCOM 21

NCOM 22

NCOM 23

NCOM 24

NCOM 25

NCOM 26

NCOM 27

NCOM 28

NCOM 29

NCOM 30

NCOM 31

NCOM 32

NCOM 33

SUBROUTINE NCOMB

PURPOSE

CALCULATES NUMBER OF POSSIBLE COMBINATIONS

USAGE

CALL NCOMB (NC,NR,NT)

DESCRIPTION OF PARAMETERS

NC = NUMBER OF COMBINATIONS

NR = NUMBER OF ITEMS TAKEN AT A TIME

NT = TOTAL NUMBER OF ITEMS

REMARKS

NONE

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED

FACTRIAL

IR = NT - NR

CALL FACTRIAL (NFACT,NT)

CALL FACTRIAL (NRFAC,NR)

CALL FACTRIAL (IRFACT,IR)

NC = NFACT / (NRFAC * IRFACT)

RETURN

END

SUBROUTINE NEWCOM (COM,M,N)

NEWC 1

NEWC 2

NEWC 3

NEWC 4

SUBROUTINE NEWCOM

NEWC 5

NEWC 6

PURPOSE

NEWC 7

NEWC 8

GENERATES M COMBINATIONS OF N INTEGERS (1,2,...,N) IN
ARRAY COM

NEWC 9

NEWC 10

USAGE

NEWC 11

NEWC 12

CALL NEWCOM (COM,M,N)

NEWC 13

DESCRIPTION OF PARAMETERS

NEWC 14

NEWC 15

COM = ARRAY CONTAINING RESULTING COMBINATION

NEWC 16

M = DIMENSION OF ARRAY COM

NEWC 17

N = LAST INTEGER OF GIVEN SERIES

NEWC 18

NEWC 19

REMARKS

NEWC 20

NEWC 21

COM MUST BE PROPERLY DIMENSIONED BY CALLING PROGRAM

NEWC 22

COM MUST CONTAIN M IN ALL THE M LOCATIONS INITIALLY

NEWC 23

$C(N,M) = \text{FACTORIAL } N / (\text{FACTORIAL } M * \text{FACTORIAL } (N-M))$

NEWC 24

SUCCESSIVE CALLS GENERATE SEQUENCE OF COMBINATIONS IN
LEXICOGRAPHIC ORDER. FURTHER CALLS CAUSE THE SEQUENCE
TO BE REPEATED

NEWC 25

NEWC 26

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED

NEWC 27

NONE

NEWC 28

NEWC 29

NEWC 30

DIMENSION COM (M)

NEWC 31

NEWC 32

TYPE INTEGER COM

NEWC 33

NEWC 34

LOCATE REGION TO BE CHANGED

NEWC 35

NEWC 36

DO 10 K = 1,M

NEWC 37

NEWC 38

KA = M - K + 1

NEWC 39

NEWC 40

KB = N - K + 1

NEWC 41

IF (COM(KA).LT.(KB) 40,10

NEWC 42

10 CONTINUE

NEWC 43

INITIATE ---- GENERATE FIRST COMBINATION

NEWC 44

NEWC 45

20 DO 30 K = 1,M

NEWC 46

NEWC 47

COM (K) = K

NEWC 48

NEWC 49

30 CONTINUE

RETURN

NEWC 50

NEWC 51

MAKE CHANGES

NEWC 52

NEWC 53

40 KB = COM (KA)

NEWC 54

NEWC 55

DO 50 K = KA,M

NEWC 56

NEWC 57

KB = KB + 1

NEWC 58

COM (K) = KB

50 CONTINUE

NEWC 59

RETURN
FILE

NEWG 57
NEWG 58
NEWG 59

SUBROUTINE STURM (DP,ICT,INDOPT,INDX11,INDX12,MAXITER,NCASE,
NP,NSEQ,NUMB,ODG,SDG,SEQ,VAL,VALUE)

STRM 1
STRM 2
STRM 3
STRM 4
STRM 5
STRM 6
STRM 7
STRM 8
STRM 9
STRM 10
STRM 11
STRM 12
STRM 13
STRM 14
STRM 15
STRM 16
STRM 17
STRM 18
STRM 19
STRM 20
STRM 21
STRM 22
STRM 23
STRM 24
STRM 25
STRM 26
STRM 27
STRM 28
STRM 29
STRM 30
STRM 31
STRM 32
STRM 33
STRM 34
STRM 35
STRM 36
STRM 37
STRM 38
STRM 39
STRM 40
STRM 41
STRM 42
STRM 43
STRM 44
STRM 45
STRM 46
STRM 47
STRM 48
STRM 49
STRM 50
STRM 51
STRM 52
STRM 53
STRM 54
STRM 55
STRM 56

SUBROUTINE STURM

PURPOSE

TRIDIAGONALIZES THE GIVEN MATRIX DP AND CARRIES OUT
STURM SEQUENCE SOLUTION FOR VARIOUS TEST EIGENVALUES
STARTING FROM UNITY AND DIVIDING BY TEN EACH TIME
STOPPING ONLY WHEN THE NUMBER OF EIGENVALUES LESS THAN
THE LAST TEST VALUE IS ZERO OR IT HAS REMAINED CONSTANT
FOR THE PAST TEN ITERATIONS. FINALLY IT GIVES THE
VALUE OF DELTA AS THE MEASURE OF TESTABILITY

USAGE

CALL STURM (DP,ICT,INDOPT,INDX11,INDX12,MAXITER,
NCASE,NP,NSEQ,NUMB,ODG,SDG,SEQ,VAL,VALUE)

DESCRIPTION OF PARAMETERS

DP = ORIGINAL SYMMETRIC POSITIVE SEMIDEFINITE
MATRIX OF DIMENSIONS NP*NP
ICT = NUMBER OF EIGENVALUES LESS THAN THE TEST VALUE
FINALLY IT IS THE VALUE OF DELTA
INDOPT = VARIABLE IF NON ZERO INTEGER INDICATES SOME
OPTION EXERCISED BY THE USER
INDX11 = VARIABLE IF NON ZERO INTEGER INDICATES USER'S
OPTION FOR PRINTOUT ON TRIDIAGONAL MATRIX AND
STURM SERIES SOLUTION (FOR EACH ITERATION)
INDX12 = VARIABLE IF NON ZERO INTEGER INDICATES USER'S
OPTION TO GET PRINTOUT ON NUMBER OF EIGENVALUES
LESS THAN TEST VALUE FOR EACH ITERATION
MAXITER = VARIABLE INDICATING MAXIMUM NUMBER OF ITERATIONS
TO BE PERFORMED FOR STURM SEQUENCE SOLUTION
IN WORST CASE
NCASE = VARIABLE INDICATING THE NUMBER OF CASE
UNDER CONSIDERATION IN CALLING PROGRAM
NP = NUMBER OF SENSITIVITY PARAMETERS
NSEQ = NUMBER OF ELEMENTS IN EACH STURM SEQUENCE
NUMB = VECTOR OF SIZE MAXITER TO CONTAIN NUMBER OF
EIGENVALUES LESS THAN TEST VALUE FOR EACH
ITERATION
ODG = ARRAY OF SIZE NP USED TO STORE ON DIAGONAL
ELEMENTS OF THE TRIDIAGONAL MATRIX
SDG = ARRAY OF SIZE NP USED TO STORE SUPER OR SUB
DIAGONAL ELEMENTS OF THE TRIDIAGONAL MATRIX
VAL = ARRAY OF SIZE MAXITER USED TO STORE THE VALUE
OF TEST EIGENVALUE FOR EACH ITERATION
VALUE = VALUE OF TEST EIGENVALUE

REMARKS

ALL ARRAYS MUST BE PROPERLY DIMENSIONED BY CALLING
PROGRAM (ECMP)
IF VALUE OF DELTA DOES NOT STABILIZE (EITHER ZERO OR
CONSTANT FOR LAST TEN ITERATIONS) UPTO THE MAXIMUM NUMBER

OF SPECIFIED ITERATIONS (MAXITER), A WARNING MESSAGE TO
THAT EFFECT IS ISSUED AND IMMEDIATE RETURN TO THE
CALLING PROGRAM (EC4P) IS CAUSED

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
SORT

METHOD

USFS HOUSEHOLDERS METHOD FOR TRIDIAGONALIZATION
AND GIVENS METHOD FOR STURM SEQUENCE COMPUTATION

DIMENSION DP(NP,NP),NUMR(MAXITER),ODG(NP),SDG(NP),
*SET(MSEN),VAL(MAXITER)

COMPUTATION OF THE TRIDIAGONAL MATRIX

NP1 = NP - 2

DO 40 J = 1, NP1

SUM = 0.0

JJ = J + 1

DO 10 K = JJ, NP

SUM = SUM + DP(K,J)**2

10 CONTINUE

S = SQRT (SUM)

IF (DP(J+1,J) .GT. 0.0) 20,30

20 S = - S

30 C = S**2 - S*DP(J+1,J)

DP(J,J+1) = DP(J,J+1) - S

DP(J+1,J) = DP(J+1,J) - S

DO 40 K = JJ, NP

ODG(K) = 0.0

DO 40 L = JJ, NP

ODG(K) = ODG(K) + DP(L,J) * DP(L,K)

40 CONTINUE

DO 50 K = JJ, NP

DO 50 L = JJ, NP

DP(K,L) = DP(K,L) - DP(K,J) * ODG(L) / C

50 CONTINUE

DO 60 K = JJ, NP

ODG(K) = 0.0

DO 60 L = JJ, NP

ODG(K) = ODG(K) + DP(K,L) * DP(L,J)

60 CONTINUE

DO 70 K = JJ, NP

DO 70 L = JJ, NP

DP(K,L) = DP(K,L) - DP(L,J) * ODG(K) / C

70 CONTINUE

STRM 570
STRM 580
STRM 590
STRM 600
STRM 610
STRM 620
STRM 630
STRM 640
STRM 650
STRM 660
STRM 670
STRM 680
STRM 690
STRM 700
STRM 710
STRM 720
STRM 730
STRM 740
STRM 750
STRM 760
STRM 770
STRM 780
STRM 790
STRM 800
STRM 810
STRM 820
STRM 830
STRM 840
STRM 850
STRM 860
STRM 870
STRM 880
STRM 890
STRM 900
STRM 910
STRM 920
STRM 930
STRM 940
STRM 950
STRM 960
STRM 970
STRM 980
STRM 990
STRM1000
STRM1010
STRM1020
STRM1030
STRM1040
STRM1050
STRM1060
STRM1070
STRM1080
STRM1090
STRM1100
STRM1110
STRM1120

```

C      DO 60 K = 1,NP
      DP(K,J) = 0.0
      DP(J,K) = 0.0
60 CONTINUE
C
      DP(J+1,J) = S
      DP(J,J+1) = S
70 CONTINUE
C
      DO 120 J = 1,NP
      DDG(J) = DP(J,J)
      IF (J-1) 100,100,110
100 SDG(1) = 0.0
      DO 10 120
110 SDG(J) = DP(J,J-1)
120 CONTINUE

```

```

C      C
C      C      OPTIONAL PRINTOUT OF THE TRIDIAGONAL MATRIX
C      C
      IF (INDOPT.NE.0) 130,150
130 IF (INDX11.NE.0) 140,150
140 WRITE (6,1000)
      WRITE (6,1010) (DDG(I),I = 1,NP)
      WRITE (6,1020)
      WRITE (6,1010) (SDG(I),I = 1,NP)

```

```

C      C
C      C      COMPUTATION OF THE STURM SEQUENCE
C      C
150 VALUE = 1.0
      J = 1
C
      DO 250 I = 1,MAXITER
      ICT = 0
      VAL(J) = VALUE
      P = 1.0
      P1 = 0.0
      SEQ(1) = P
C
      DO 140 L = 1,NP
      P1 = (DDG(L) - VALUE)*P - SDG(L)**2*P1
      Q = P
      P = P1
      P1 = 0
      SEQ(L+1) = P
140 CONTINUE

```

```

C      C
C      C      OPTIONAL PRINTOUT OF STURM SEQUENCE SOLUTION
C      C
      IF (INDOPT.NE.0) 170,190
170 IF (INDX11.NE.0) 180,190
180 WRITE (6,1030) VALUE
      WRITE (6,1010) (SEQ(K),K = 1,NSEQ)

```

```

C      C
C      C      COMPUTATION OF DELTA
C      C

```

STRM1130
 STRM1140
 STRM1150
 STRM1160
 STRM117
 STRM1180
 STRM1190
 STRM1200
 STRM1210
 STRM1220
 STRM123
 STRM1240
 STRM1250
 STRM1260
 STRM1270
 STRM1280
 STRM129
 STRM1300
 STRM1310
 STRM1320
 STRM1330
 STRM1340
 STRM135
 STRM136
 STRM137
 STRM138
 STRM139
 STRM140
 STRM141
 STRM142
 STRM143
 STRM144
 STRM145
 STRM146
 STRM147
 STRM148
 STRM149
 STRM150
 STRM151
 STRM152
 STRM153
 STRM154
 STRM155
 STRM156
 STRM157
 STRM158
 STRM159
 STRM160
 STRM161
 STRM162
 STRM163
 STRM164
 STRM165
 STRM166
 STRM167
 STRM168
 STRM169
 STRM170


```

140 11 210 K = 1,NP
    IF (SQR(1+1) * SFQ(K)) 200,210,210
200 ICT = ICT + 1
210 CONTINUE

```

```

    NUMB(J) = ICT
    IF (ICT.EQ.0) 250,220
220 IF (J.GE.10) 230,240
230 IF (NUMB(J)*NUMB(J-1)*NUMB(J-2)*NUMB(J-3)*NUMB(J-4)*NUMB(J-5)*
    *NUMB(J-6)*NUMB(J-7)*NUMB(J-8)*NUMB(J-9).EQ.NUMB(J)**10) 260,240
240 VALUE = VALUE / 10.0
    J = J + 1
250 CONTINUE

```

```

    WRITE (6,1040) NCASE
    GO TO 300

```

OPTIONAL PRINTOUT FOR EACH EIGENVALUE COMPARISON

```

260 IF (INDOPT.NE.0) 270,300
270 IF (INDX12.NE.0) 280,300
280 WRITE (6,1050)

```

```

DO 290 I = 1,J
    WRITE (6,1060) VAL(I), NUMB(I)
290 CONTINUE

```

FORMAT STATEMENTS

```

1000 FORMAT (54H1 THE DIAGONAL ELEMENTS OF THE TRIDIAGONAL MATRIX ARE/)
1010 FORMAT (1H0,10X,E13.3)
1020 FORMAT (66H1 THE SUPER AND SUB DIAGONAL ELEMENTS OF THE TRIDIAGON
    *AL MATRIX ARE/)
1030 FORMAT (54H1 THE STURM SEQUENCE SOLUTION FOR A TEST EIGENVALUE OF,
    *2X,E13.3,2X,2HIS/)
1040 FORMAT (85H1 ***WARNING MESSAGE** VALUE OF DELTA DOES NOT STABILIZE
    *FOR MAXITER SPECIFIED IN CASE,2X,13,2X,26HAND MAY NOT BE RELIED UP
    *ON/)
1050 FORMAT (1H1)
1060 FORMAT (37H0 THE NUMBER OF EIGENVALUES LESS THAN,
    *2X,E13.3,2X,2HIS,2X,13)
300 RETURN
END

```

STRM1690
 STRM1700
 STRM1710
 STRM1720
 STRM1730
 STRM1740
 STRM1750
 STRM1760
 STRM1770
 STRM1780
 STRM1790
 STRM1800
 STRM1810
 STRM1820
 STRM1830
 STRM1840
 STRM1850
 STRM1860
 STRM1870
 STRM1880
 STRM1890
 STRM1900
 STRM1910
 STRM1920
 STRM1930
 STRM1940
 STRM1950
 STRM1960
 STRM1970
 STRM1980
 STRM1990
 STRM2000
 STRM2010
 STRM2020
 STRM2030
 STRM2040
 STRM2050
 STRM2060
 STRM2070
 STRM2080
 STRM2090
 STRM2100
 STRM2110
 STRM2120

SUBROUTINE SUBM (X,IX,JX,Y,IY,JY,Z,IZ,JZ)

SUBROUTINE SUBM

PURPOSE

SUBTRACTS MATRIX Y FROM MATRIX X TO PRODUCE MATRIX Z

USAGE

CALL SUBM (X,IX,JX,Y,IY,JY,Z,IZ,JZ)

DESCRIPTION OF PARAMETERS

X = FIRST MATRIX

IX AND JX = DIMENSIONS OF MATRIX X

Y = SECOND MATRIX TO BE SUBTRACTED FROM FIRST

IY AND JY = DIMENSIONS OF MATRIX Y

Z = RESULTING DIFFERENCE MATRIX

IZ AND JZ = DIMENSIONS OF MATRIX Z

REMARKS

MATRICES MUST BE PROPERLY DIMENSIONED BY CALLING PROGRAM
TERMINATES JOB IF DIMENSIONS DO NOT MATCH

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED

NONE

DIMENSION X (IX,JX), Y (IY,JY), Z (IZ,JZ)

IF (IX.EQ.IY) 10,40

10 IF (JX.EQ.JY) 20,40

20 IZ = IX

JZ = JX

DO 30 I = 1,IZ

DO 30 J = 1,JZ

Z(I,J) = X(I,J) - Y(I,J)

30 CONTINUE

RETURN

40 WRITE (5,1000)

1000 FORMAT (50H **ERROR** MATRICES NOT COMPATIBLE FOR SUBTRACTION)

END

SUBROUTINE TRAM (X,IX,JX,Y,IY,JY)

TRAM 1

TRAM 2

TRAM 3

TRAM 4

TRAM 5

TRAM 6

TRAM 7

TRAM 8

TRAM 9

TRAM 10

TRAM 11

TRAM 12

TRAM 13

TRAM 14

TRAM 15

TRAM 16

TRAM 17

TRAM 18

TRAM 19

TRAM 20

TRAM 21

TRAM 22

TRAM 23

TRAM 24

TRAM 25

TRAM 26

TRAM 27

TRAM 28

TRAM 29

TRAM 30

TRAM 31

TRAM 32

TRAM 33

TRAM 34

TRAM 35

TRAM 36

TRAM 37

SUBROUTINE TRAM

PURPOSE

COMPUTES TRANSPOSE OF MATRIX X

USAGE

CALL TRAM (X,IX,JX,Y,IY,JY)

DESCRIPTION OF PARAMETERS

X = MATRIX WHOSE TRANSPOSE IS TO BE COMPUTED

IX AND JX = DIMENSIONS OF MATRIX X

Y = RESULTING MATRIX WHICH IS TRANSPOSE OF X

IY AND JY = DIMENSIONS OF MATRIX Y

REMARKS

MATRICES MUST BE PROPERLY DIMENSIONED BY CALLING PROGRAM

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED

NONE

DIMENSION X(IX,JX) , Y(JX,IX)

IY = JX

JY = IX

DO 10 I = 1,IY

DO 10 J = 1,JY

Y(I,J) = X(J,I)

10 CONTINUE

RETURN

END

SUBROUTINE VEC (X,IX,IY,Y,IJ)

VEC 11

VEC 21

VEC 31

VEC 41

SUBROUTINE VEC

VEC 51

VEC 61

PURPOSE

VEC 71

TRANSFORMS A MATRIX INTO A SINGLE COLUMN VECTOR

VEC 81

VEC 91

USAGE

VEC 101

CALL VEC (X,IX,IY,Y,IJ)

VEC 111

VEC 121

DESCRIPTION OF PARAMETERS

VEC 131

X = ORIGINAL MATRIX OF SIZE IX*IX

VEC 141

IX AND IY = DIMENSIONS OF MATRIX X

VEC 151

Y = RESULTING COLUMN VECTOR OF SIZE IJ

VEC 161

IJ = DIMENSION OF THE COLUMN VECTOR Y

VEC 171

VEC 181

REMARKS

VEC 191

X AND Y MUST BE PROPERLY DIMENSIONED BY CALLING PROGRAM

VEC 201

VEC 211

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED

VEC 221

NONE

VEC 231

VEC 241

VEC 251

DIMENSION X(IX,IY),Y(IJ)

VEC 261

K = 0

VEC 271

VEC 281

VEC 291

DO 10 I = 1,IY

VEC 301

DO 10 J = 1,IX

VEC 311

K = K + 1

VEC 321

Y (K) = X (J,I)

VEC 331

10 CONTINUE

VEC 341

VEC 351

RETURN

VEC 361

END

VEC 371

SUBROUTINE ZERM (X,IX,JX)

ZERM 10

ZERM 20

ZERM 30

ZERM 40

ZERM 50

ZERM 60

ZERM 70

ZERM 80

ZERM 90

ZERM 100

ZERM 110

ZERM 120

ZERM 130

ZERM 140

ZERM 150

ZERM 160

ZERM 170

ZERM 180

ZERM 190

ZERM 200

ZERM 210

ZERM 220

ZERM 230

ZERM 240

ZERM 250

ZERM 260

ZERM 270

ZERM 280

ZERM 290

ZERM 300

ZERM 310

ZERM 320

ZERM 330

SUBROUTINE ZERM

PURPOSE

PRODUCES A MATRIX WITH ALL ENTRIES ZERO

USAGE

CALL ZERM (X,IX,JX)

DESCRIPTION OF PARAMETERS

X = MATRIX WITH ALL ENTRIES ZERO

IX AND JX = DIMENSIONS OF MATRIX X

REMARKS

MATRIX MUST BE PROPERLY DIMENSIONED BY CALLING PROGRAM

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED

NONE

DIMENSION X (IX,JX)

DO 10 I = 1,IX

DO 10 J = 1,JX

X (I,J) = 0.0

10 CONTINUE

RETURN

END

2.8 An Example Run on Package

2.8.1 Model Set Up

An example, consider the amplifier circuit shown in Figure 2.

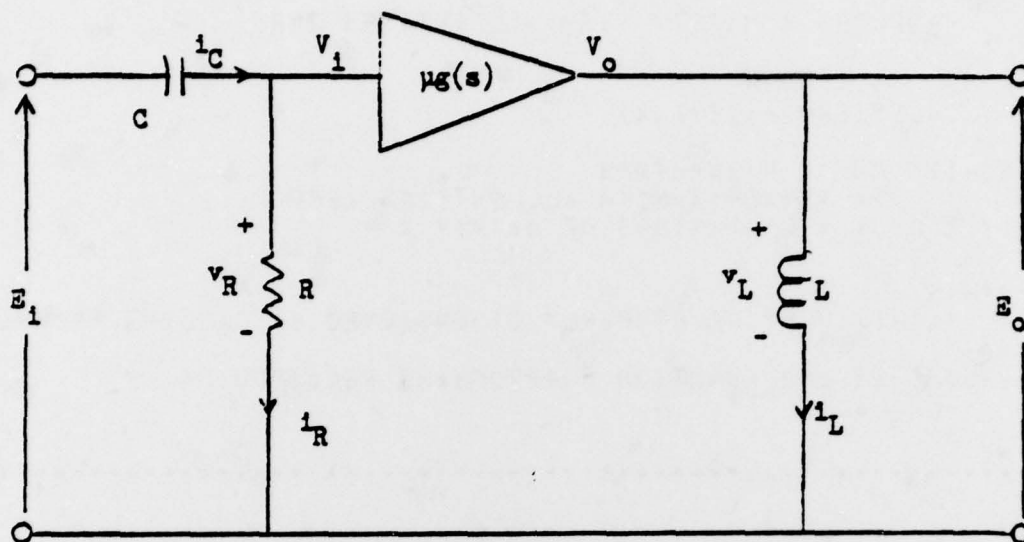


Figure 2. RC-Coupled Amplifier with an Inductive Load

Assuming four outputs available (for comparison purposes), and a single input, a set of connection equations for the circuit are given in equation 2.8A and the corresponding set of component equations are given in equation 2.8B.

$$\begin{bmatrix} v_i \\ v_L \\ i_C \\ v_R \\ \vdots \\ E_0 \\ i_0 \\ i_R \\ v_i \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \\ \hline 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v_0 \\ i_L \\ v_C \\ i_R \\ \vdots \\ E_i \end{bmatrix} \quad (2.8A)$$

$$\begin{bmatrix} v_0 \\ i_L \\ v_C \\ i_R \end{bmatrix} = \begin{bmatrix} \mu(r)g(s) & 0 & 0 & 0 \\ 0 & 1/L(r)s & 0 & 0 \\ 0 & 0 & 1/C(r)s & 0 \\ 0 & 0 & 0 & 1/R(r) \end{bmatrix} \begin{bmatrix} v_i \\ v_L \\ i_C \\ v_R \end{bmatrix} \quad (2.8B)$$

where r denotes the parameter subject to variation which in our example will be μ , R , L and C themselves.

As δ is constant almost everywhere [3] and is thus independent of the actual component values, for computational purposes one can take the nominal value of all the four components to be unity. Also 15 real frequencies with a starting frequency of 1 and a step size of unity will serve the purpose. The maximum number of possible cases (MAXCAS) for various combinations of test inputs and outputs is 15. The costs are assigned to inputs and outputs as per equation 2.8C.

Inputs			Outputs		
Number Designation	Type Designation	Cost	Number Designation	Type Designation	Cost
1	E_1	1	1	V_i	1
			2	V_L	2
			3	i_C	3
			4	V_R	1

(2.8C)

The Data Cards set up for this example are shown in Figure 3. A printout of the three user supplied program is given in Section 2.8.2. Section 2.8.3 contains a copy of the standard output.

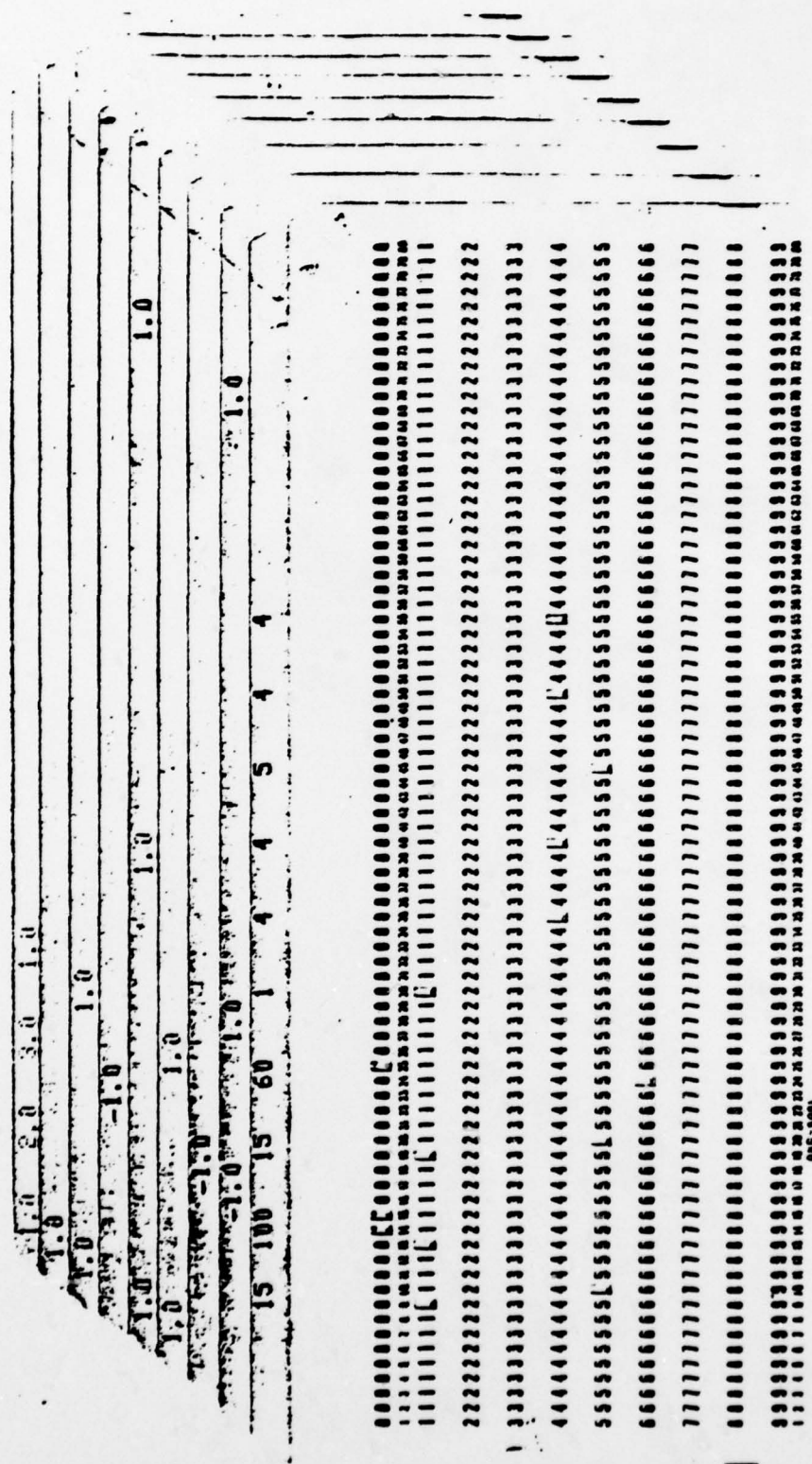


Figure 3. Data Cards Set-Up.

2.8.2 User Supplied Subroutines

RESEARCH PAGE NOT FILLED
BLACK

```

PROGRAM TPCORICE
DIMENSION A(4,4),AL11(4,4),AL12(4,1),AL21(4,4),AL22(4,1),H(4,4),
+ C(4,4),CCOF(60,4),COF(15,4,4),COL(4),COSTO(4),COSTI(1),D(4,4),
+ DIFZ(4,4),DP(4,4),E(4,4),F(4,4),G(4,4),H(4,4),LCOMI(15),LCOMO(15),
+ NUMH(100),ODG(4),P(4,4),R(4,1),SDG(4),SENS(15,4,4),SEQ(5),SYSM
+ (4,1),T(4,1),TCCOF(4,60),VAL(100),VRF(10),VWF(10),W(4,1),XI(4,4),
+ Z(4,4)
READ (5,1000) INDOPT,MAXCAS,MAXITER,NF,NFVEC,NI,NO,NP,NSFQ,NVEC,NZ
CALL FCMP (A,AL11,AL12,AL21,AL22,H,C,CCOF,COF,COL,
+ COSTI,COSTO,D,DIFZ,DP,E,F,G,H,INDOPT,LCOMI,LCOMO,MAXCAS,MAXITER,
+ NF,NFVEC,NI,NO,NP,NSFQ,NUMH,NVEC,NZ,ODG,P,R,
+ SDG,SENS,SEQ,SYSM,T,TCCOF,VAL,VRF,VWF,W,XI,Z )
1000 FORMAT (11I5)
END

```

```

SUBROUTINE COMPT (NZ,S,Z)
  DIMENSION Z (NZ,NZ)
  Z (1,1) = (1.0) / ((S+1.0) * (S+2.0))
  Z (2,2) = 1.0 / S
  Z (3,3) = 1.0 / S
  Z (4,4) = 1.0
  RETURN
  END

```

```

SUBROUTINE SENSITVY (DIFZ,KS,NZ,S)
  DIMENSION DIFZ (NZ,NZ)
  GO TO (1,2,3,4),KS
1  DIFZ (1,1) = 1.0 / ((S+1.0) * (S+2.0))
  GO TO 100
2  DIFZ (2,2) = - 1.0 / S
  GO TO 100
3  DIFZ (3,3) = - 1.0 / S
  GO TO 100
4  DIFZ (4,4) = - 1.0
100 RETURN
END

```


2.8.3 Standard Output

1	1	1	2.000E 00	2	1.000E-13
2	1	2	3.000E 00	2	1.000E-14
3	1	3	4.000E 00	2	1.000E-10
4	1	4	2.000E 00	3	1.000E-10
5	1	1	4.000E 00	1	1.000E-13
		2			
6	1	1	5.000E 00	1	1.000E-11
		3			
7	1	1	3.000E 00	2	1.000E-11
		4			
8	1	2	6.000E 00	0	1.000E-11
		3			
9	1	2	4.000E 00	2	1.000E-11
		4			
10	1	3	5.000E 00	2	1.000E-10
		4			
11	1	1	7.000E 00	0	1.000E-03
		2			
		3			
12	1	1	5.000E 00	1	1.000E-12
		2			
		4			
13	1	1	6.000E 00	1	1.000E-11
		3			
		4			
14	1	2	7.000E 00	0	1.000E-11
		3			
		4			
15	1	1	8.000E 00	0	1.000E-03
		2			
		3			
		4			

RESEARCH AND ANALYSIS
LABORATORY

References

1. Sen, N. and Saeks, R., "A Measure of Testability and its Applications to Test Point Selection - Theory", Proc. of 20th Midwest Symposium on Circuits and Systems, Lubbock, Texas, August 1977.
2. Sen, N. and Saeks R., "A Measure of Testability and its Applications to Test Point Selection - Computation", Proc. of 1977 IEEE Inter. Automatic Testing Conference, Hyannis, Mass., November 1977.
3. Sen, N., M. S. Thesis, Texas Tech University, 1977.
4. Ransom, M.N., "A Functional Approach to the Connections of a Large-Scale Dynamical System", Ph.D., Thesis, Univ. of Notre Dame, 1973.